



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

TUGAS AKHIR - KI141502

## **RANCANG BANGUN APLIKASI WEB LELANG *ONLINE* (E-AUCTION) BERBASIS KERANGKA KERJA LARAVEL**

**RONAULI SILVA NATALENSIS SIDABUKKE**  
NRP 5113100142

Dosen Pembimbing I  
Rully Soelaiman, S.Kom, M.Kom

Dosen Pembimbing II  
Rizky Januar Akbar, S.Kom., M.Eng

**JURUSAN TEKNIK INFORMATIKA**  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember  
Surabaya, 2017

*(Halaman ini sengaja dikosongkan)*

**TUGAS AKHIR - KI141502**

**RANCANG BANGUN APLIKASI *WEB LELANG ONLINE*  
(*E-AUCTION*) BERBASIS KERANGKA KERJA LARAVEL**

**RONAULI SILVA NATALENSIS SIDABUKKE**  
**NRP 5113100142**

**Dosen Pembimbing I**  
**Rully Soelaiman, S.Kom, M.Kom**

**Dosen Pembimbing II**  
**Rizky Januar Akbar, S.Kom., M.Eng**

**JURUSAN TEKNIK INFORMATIKA**  
**Fakultas Teknologi Informasi**  
**Institut Teknologi Sepuluh Nopember**  
**Surabaya, 2017**

*(Halaman ini sengaja dikosongkan)*

UNDERGRADUATE THESIS - KI141502

**E-AUCTION WEB APPLICATION DESIGN AND  
IMPLEMENTATION BASED ON LARAVEL FRAMEWORK**

RONAULI SILVA NATALENSIS SIDABUKKE  
NRP 5113100142

Supervisor I  
Rully Soelaiman, S.Kom, M.Kom

Supervisor II  
Rizky Januar Akbar, S.Kom., M.Eng

Department of INFORMATICS  
Faculty of Information Technology  
Institut Teknologi Sepuluh Nopember  
Surabaya, 2017

*(Halaman ini sengaja dikosongkan)*

## **LEMBAR PENGESAHAN**

### **RANCANG BANGUN APLIKASI *WEB* LELANG *ONLINE* (*E-AUCTION*) BERBASIS KERANGKA KERJA LARAVEL**

#### **TUGAS AKHIR**

Diajukan Guna Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada

Bidang Studi Algoritma Pemrograman  
Program Studi S1 Jurusan Teknik Informatika  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember

Oleh:

**RONAULI SILVA NATALENSIS SIDABUKKE**  
**NRP: 5113100142**

Disetujui oleh Dosen Pembimbing Tugas Akhir :

Rully Soelaiman, S.Kom, M.Kom

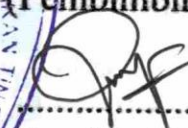
NIP: 197002131994021001



(Pembimbing 1)

Rizky Januar Akbar, S.Kom, M.Eng

NIP: 198701032014041001



(Pembimbing 2)

**SURABAYA**

**Juni 2017**

*(Halaman ini sengaja dikosongkan)*



# **RANCANG BANGUN APLIKASI *WEB* LELANG *ONLINE* (*E-AUCTION*) BERBASIS KERANGKA KERJA LARAVEL**

**Nama** : RONAULI SILVA NATALENSIS  
SIDABUKKE  
**NRP** : 5113100142  
**Jurusan** : Teknik Informatika FTIf  
**Pembimbing I** : Rully Soelaiman, S.Kom, M.Kom  
**Pembimbing II** : Rizky Januar Akbar, S.Kom., M.Eng

## **Abstrak**

*E-commerce adalah kombinasi antara dunia digital dan transaksi lelang. Di Indonesia, seiring terjadi peningkatan jumlah pengguna internet dan menjamurnya bisnis online atau sering disebut online shop. Salah satu jenis transaksi adalah lelang, yaitu metode jual beli yang mengintegrasikan mekanisme lelang dengan internet.*

*Dalam interaksi antara pelaku lelang online (penjual dan pembeli) pasti terjadi kegagalan/ketidakpuasan dalam transaksi lelang online. Berangkat dari penelitian "Online auction service failures in Taiwan: Typologies and recovery strategies" yang membahas mengenai analisa kesalahan dan strategi lewat survey terhadap pengguna aplikasi lelang online di Taiwan, aplikasi lelang online ini dibangun disertai dengan tambahan fitur maupun saran dari penelitian tersebut.*

*Analisa aplikasi e-commerce yang umum digunakan di Indonesia juga dilakukan, baik dari aspek user experience maupun alur transaksi agar lebih sesuai dengan transaksi jual-beli online yang umum di Indonesia. Dengan tugas akhir ini, diharapkan kegagalan dalam transaksi online dapat diperbaiki dan membuka peluang lelang online untuk meramaikan industri e-commerce di Indonesia.*

***Kata-Kunci:*** *lelang online, strategi*

# **E-AUCTION WEB APPLICATION DESIGN AND IMPLEMENTATION BASED ON LARAVEL FRAMEWORK**

**Name : RONAULI SILVA NATALENSIS  
SIDABUKKE**  
**NRP : 5113100142**  
**Major : Informatics FTIf**  
**Supervisor I : Rully Soelaiman, S.Kom, M.Kom**  
**Supervisor II : Rizky Januar Akbar, S.Kom., M.Eng**

## **Abstract**

*E-commerce industry is growing rapidly in Indonesia, along with the increasing number of internet users and number of online shops is also growing. One of e-commerce type is online auction, a buy and sell method that integrates auction mechanism and the internet.*

*In the interaction between online auction actors (buyers and sellers), inevitable failure/dissatisfaction of online auction transactions sometimes found. Started by analysing research of "Online auction service failures in Taiwan: Typologies and recovery strategies", through users survey, author want to build online auction application along with additional added values and suggestions from the research.*

*Analysis and observation are also conducted, especially in user experience aspects, design and transaction flow local e-commerce platforms that are commonly used in Indonesia, in purpose to make the application suits Indonesian's users better. In hope that this application could reduce/prevent the expected failures in online transactions and open up online auction opportunity to enliven the e-commerce industry in Indonesia.*

**Keyword:** *online auction, typologies and strategies*

*(Halaman ini sengaja dikosongkan)*

## KATA PENGANTAR

Puji Syukur kepada Tuhan yang Maha Esa, atas berkatNya penulis dapat menyelesaikan buku berjudul **Rancang Bangun Aplikasi web Lelang Online (E-Auction) Berbasis Kerangka Kerja Laravel**.

Selain itu, pada kesempatan ini penulis menghaturkan terima kasih sebesar-besarnya kepada pihak-pihak yang tanpa mereka, penulis tidak akan dapat menyelesaikan buku ini:

1. **Daddy Jesus** - atas segala berkat, limpahan karunia, kesempatan dan rancangan jalanNya-lah penulis masih diberi nafas kehidupan, waktu, tenaga dan pikiran untuk menyelesaikan buku ini. *Thank you, Big Daddy*.
2. **Papa dan Mama** yang selalu menguatkan, menasehati, dan luar biasa sabar dalam mengingatkan penulis agar tidak lupa menjaga kesehatan dan tidak lupa ke gereja selama masa studi.
3. **Yth Bapak Rully Soelaiman** yang mengajarkan penulis *how to think scientifically* juga bimbingan, nasehat, saran dan memberikan penulis sisi pemikiran dan perspektif lain terhadap setiap masalah; dan  
**Yth Bapak Rizky Januar Akbar** sebagai dosen pembimbing yang memberi bimbingan, saran teknis dan administratif, diskusi dan pemecahan masalah dalam pembuatan dan penulisan buku tugas akhir.
4. **Keluarga XL Future Leader Scholarship Camp Batch 5 & KSE ITS** yang telah menyadarkan, memberikan semangat dan inspirasi untuk terus melanjutkan tugas akhir di saat penulis kehilangan semangat.
5. **Teman-teman baik penulis:** Gideon, Bilflash, Fablius, Luthfie, Gresela, Ruth Mandha, Rifat, Petrus, Lusiana Eka, Petrus yang sudah rela menjadi *support system* penulis.
6. **Keluarga Admin Lab. Pemrograman (2014 - 2017)**, yang bersama-sama belajar dalam lingkungan lab;
7. **Keluarga Pengpro Furions dan HMTc Optimasi 2016** , yang mengajarkan penulis tentang cara berorganisasi, cara

berbicara di depan publik, dan banyak lagi;

8. **Keluarga Alumni Budi Mulia Siantar-Surabaya angkatan 2013**

9. Serta semua pihak yang tidak tertulis, baik yang membantu dalam proses pengujian, membantu memikirkan saat ada masalah, dan lainnya yang telah turut membantu penulis dalam menyelesaikan Tugas Akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan. Oleh karena itu, penulis berharap kritik dan saran dari pembaca sekalian untuk memperbaiki buku ini ke depannya.

Surabaya, Juli 2017

Ronauli Silva N. Sidabukke

## DAFTAR ISI

<b>ABSTRAK</b> .....	<b>iii</b>
<b>ABSTRACT</b> .....	<b>v</b>
<b>KATA PENGANTAR</b> .....	<b>vii</b>
<b>DAFTAR ISI</b> .....	<b>ix</b>
<b>DAFTAR TABEL</b> .....	<b>xiii</b>
<b>DAFTAR GAMBAR</b> .....	<b>xvii</b>
<b>DAFTAR KODE SUMBER</b> .....	<b>xxi</b>
<b>BAB I PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan .....	2
1.5 Metodologi .....	3
1.6 Sistematika Penulisan.....	3
<b>BAB II LANDASAN TEORI</b> .....	<b>5</b>
2.1 Amazon Web Service .....	5
2.2 <i>Concurrency</i> .....	5
2.3 <i>Data Growth</i> .....	6
2.4 <i>JSON Web Token</i> .....	6
2.5 Laravel.....	7
2.6 Laravel Dusk .....	7
2.7 Lelang.....	8
2.8 Lelang Daring/ <i>Lelang Online</i> .....	8
2.9 MongoDB.....	9
2.10 Node.js .....	9
2.11 NoSQL .....	9
2.12 npm / <i>Node Package Manager</i> .....	10
2.13 PostgreSQL.....	10
2.14 <i>Repository Pattern</i> .....	11
2.15 SendGrid .....	11
2.16 <i>Service Worker</i> .....	12
2.17 SMTP / <i>Simple Mail Transfer Protocol</i> .....	12
2.18 Socket.io .....	12
2.19 <i>Test Script</i> .....	13

2.20 <i>Vendu Reglement</i> .....	13
2.21 <i>Vue.js / Vue</i> .....	13
2.22 <i>Whitelist</i> .....	14
<b>BAB III ANALISA DAN PERANCANGAN SISTEM.....</b>	<b>15</b>
3.1 Analisa Sistem.....	15
3.1.1 Deskripsi Umum Aplikasi .....	15
3.1.2 Analisa Paper Rujukan .....	16
3.1.3 <i>Bussiness Aspects of Software Engineering</i> .....	18
3.1.4 <i>Technical Analysis</i> .....	20
3.1.5 Analisa Penulis .....	21
3.1.6 Spesifikasi Kebutuhan Non-Fungsional.....	28
3.1.7 Tugas dan Hak Akses Aktor .....	30
3.2 Perancangan Sistem .....	31
3.2.1 Spesifikasi Kasus Penggunaan .....	31
3.2.2 Identifikasi Komponen Fundamental .....	57
3.2.3 <i>Technology Options</i> .....	57
3.2.4 Arsitektur Perangkat Lunak .....	61
3.2.5 Struktur Aplikasi.....	63
3.2.6 Perancangan <i>Database</i> .....	65
3.2.7 Kamus Data <i>Database</i> Transaksional.....	68
3.2.8 Kamus Data <i>Database</i> Non-Transaksional .....	78
<b>BAB IV IMPLEMENTASI .....</b>	<b>81</b>
4.1 Implementasi Perangkat Keras/ <i>Deployment</i> .....	81
4.2 Implementasi Perangkat Lunak .....	82
4.2.1 Strategi <i>Deployment</i> NGINX .....	82
4.2.2 Strategi <i>Deployment</i> VUE.JS .....	83
4.2.3 Strategi <i>Whitelisting</i> pada SENDGRID.....	87
4.3 Implementasi Antarmuka / <i>User Interface</i> .....	89
4.3.1 Antarmuka Halaman Registrasi .....	89
4.3.2 Implementasi Halaman <i>Login</i> .....	91
4.3.3 Melihat Daftar Barang yang Dillang .....	93
4.3.4 Menawar Barang.....	96
4.3.5 Mendaftarkan Barang untuk Dilelang .....	100



4.3.6	Memperbarui Barang .....	106
4.3.7	Melihat Barang yang Didaftarkan .....	115
4.3.8	Menambahkan Review .....	116
4.3.9	Mengirimkan Pesan .....	121
4.3.10	Melihat Kotak Pesan .....	127
4.3.11	Memasukkan Kupon .....	132
4.3.12	Melihat Daftar Pengguna .....	135
4.3.13	Menambahkan Kupon .....	136
4.3.14	Melihat Daftar Kupon .....	138
4.3.15	Memperbarui Kupon .....	139
<b>BAB V</b>	<b>PENGUJIAN DAN EVALUASI .....</b>	<b>143</b>
5.1	Pengujian .....	143
5.1.1	Metode Pengujian.....	143
5.1.2	Pengujian Fungsionalitas .....	144
5.1.3	Pengujian Kecepatan.....	155
5.1.4	<i>User Experience Assesment</i> .....	157
5.1.5	<i>Maintainability Assesment</i> .....	159
5.2	Evaluasi .....	160
5.2.1	Pendekatan Hukum Perlindungan Konsumen ...	161
5.2.2	Evaluasi Pengujian <i>User Experience</i> .....	166
5.2.3	Evaluasi Pengujian Kecepatan.....	168
5.2.4	Evaluasi Pengujian <i>Maintainability</i> .....	169
5.2.5	<i>Summary</i> Evaluasi .....	171
<b>BAB VI</b>	<b>PENUTUP.....</b>	<b>173</b>
6.1	Kesimpulan.....	173
6.2	Saran .....	173
<b>DAFTAR PUSTAKA</b> .....		<b>175</b>
<b>BAB A</b>	<b>Kode Sumber .....</b>	<b>179</b>
<b>BAB B</b>	<b>Kuisisioner Pengujian <i>User Experience</i> .....</b>	<b>189</b>
<b>BAB C</b>	<b>Kuisisioner Pengujian <i>Maintainability</i>.....</b>	<b>201</b>
<b>BIODATA PENULIS</b> .....		<b>209</b>

*(Halaman ini sengaja dikosongkan)*

## DAFTAR TABEL

Tabel 3.1 Kebutuhan Fungsional Aplikasi Lelang <i>Online</i> ....	27
Tabel 3.2 Kebutuhan Non-Fungsional Aplikasi Lelang <i>Online</i> .....	28
Tabel 3.3 Identifikasi Aktor .....	30
Tabel 3.4 Detail Tugas dan Hak Akses .....	31
Tabel 3.5 Tabel Kasus Penggunaan.....	32
Tabel 3.6 Spesifikasi Kasus Penggunaan Registrasi.....	34
Tabel 3.7 Spesifikasi Kasus Penggunaan <i>Login</i> .....	35
Tabel 3.8 Spesifikasi Kasus Penggunaan: Konfirmasi <i>Email</i>	36
Tabel 3.9 Spesifikasi Kasus Penggunaan: Melihat Barang yang Dilelang .....	38
Tabel 3.10 Spesifikasi Kasus Penggunaan: Mencari Barang Lelang.....	39
Tabel 3.11 Spesifikasi Kasus Penggunaan: Mencari Barang Lelang.....	40
Tabel 3.12 Spesifikasi Kasus Penggunaan: Melihat Riwayat Penawaran Lelang Barang .....	41
Tabel 3.13 Spesifikasi Kasus Penggunaan: Mendaftarkan Barang Lelang .....	43
Tabel 3.14 Spesifikasi Kasus Penggunaan: Mendaftarkan Barang Lelang .....	44
Tabel 3.15 Spesifikasi Kasus Penggunaan: Melihat Barang yang Pernah Didaftarkan.....	45
Tabel 3.16 Spesifikasi Kasus Penggunaan: Melihat Riwayat Penawaran Harga.....	46
Tabel 3.17 Spesifikasi Kasus Penggunaan: Melihat <i>Review</i> Pengguna.....	48
Tabel 3.18 Spesifikasi Kasus Penggunaan: Memberikan <i>Review</i> Pengguna .....	49
Tabel 3.19 Spesifikasi Kasus Penggunaan: Mengirimkan Pesan .....	50
Tabel 3.20 Spesifikasi Kasus Penggunaan: Melihat & Membaca Pesan .....	51

Tabel 3.21	Spesifikasi Kasus Penggunaan: Menambahkan Kupon .....	53
Tabel 3.22	Spesifikasi Kasus Penggunaan: Mendaftarkan Barang Lelang .....	54
Tabel 3.23	Spesifikasi Kasus Penggunaan: Menambahkan Kupon .....	55
Tabel 3.24	Spesifikasi Kasus Penggunaan: Mendaftarkan Barang Lelang .....	56
Tabel 3.25	Kamus Data Tabel <i>administrators</i> .....	68
Tabel 3.26	Kamus Data Tabel <i>bids</i> .....	68
Tabel 3.27	Kamus Data Tabel <i>categories</i> .....	69
Tabel 3.28	Kamus Data Tabel <i>coupon</i> .....	70
Tabel 3.29	Kamus Data Tabel <i>coupon_usages</i> .....	71
Tabel 3.30	Kamus Data Tabel <i>favorites</i> .....	71
Tabel 3.31	Kamus Data Tabel <i>issues</i> .....	72
Tabel 3.32	Kamus Data Tabel <i>issuetypes</i> .....	72
Tabel 3.33	Kamus Data Tabel <i>items</i> .....	73
Tabel 3.34	Kamus Data Tabel <i>ratingauctioneers</i> .....	74
Tabel 3.35	Kamus Data Tabel <i>ratingbidders</i> .....	75
Tabel 3.36	Kamus Data Tabel <i>ratinglogs</i> .....	75
Tabel 3.37	Kamus Data Tabel <i>userverifications</i> .....	76
Tabel 3.38	Kamus Data Tabel <i>userissues</i> .....	77
Tabel 3.39	Kamus Data Tabel <i>users</i> .....	77
Tabel 3.40	Kamus Data <i>Collection userchat</i> .....	78
Tabel 3.41	Kamus Data <i>Collection chatroom</i> .....	79
Tabel 3.42	Kamus Data <i>Collection itemimage</i> .....	79
Tabel 5.1	Pengujian Fungsionalitas Fitur Manajemen Akun .	144
Tabel 5.2	Pengujian Fungsionalitas Fitur Manajemen Penawaran .....	146
Tabel 5.3	Pengujian Fungsionalitas Fitur Manajemen Barang Lelang .....	148

Tabel 5.4	Pengujian Fungsionalitas Fitur Manajemen Interaksi Antarpengguna .....	150
Tabel 5.5	Pengujian Fungsionalitas Fitur Manajemen Kupon .....	152
Tabel 5.6	Pengujian Fungsionalitas Fitur Manajemen Interaksi Antarpengguna .....	156
Tabel 5.7	Rekapitulasi Hasil Pengujian Pengguna Aplikasi Lelang <i>Online</i> .....	158
Tabel 5.8	Rekapitulasi Pengujian <i>Maintainability</i> pada Aspek Kode Sumber .....	160
Tabel 5.9	Rekapitulasi Pengujian <i>Maintainability</i> pada Aspek Dokumentasi .....	160
Tabel 5.10	Rekapitulasi Perbedaan Hasil Pengujian Pengguna Aplikasi Lelang <i>Online</i> .....	167
Tabel 5.11	Tabel Evaluasi Hasil Pengujian <i>Maintainability</i> ...	170
Tabel 5.12	Evaluasi Keseluruhan .....	171
Tabel 3.1	Kuisisioner Pengujian <i>Maintainability</i> Responden 1	201
Tabel 3.2	Kuisisioner Pengujian <i>Maintainability</i> Responden 2	202
Tabel 3.3	Kuisisioner Pengujian <i>Maintainability</i> Responden 3	202
Tabel 3.4	Kuisisioner Pengujian <i>Maintainability</i> Responden 4	203
Tabel 3.5	Kuisisioner Pengujian <i>Maintainability</i> Responden 5	204
Tabel 3.6	Kuisisioner Pengujian <i>Maintainability</i> Responden 6	205
Tabel 3.7	Kuisisioner Pengujian <i>Maintainability</i> Responden 7	206
Tabel 3.8	Kuisisioner Pengujian <i>Maintainability</i> Responden 8	207

*(Halaman ini sengaja dikosongkan)*

## DAFTAR GAMBAR

Gambar 3.1	Fatalitas Kegagalan dalam Aplikasi Lelang <i>Online</i> dan Kepuasan Perbaikan Pelayanan dan <i>Repeat Purchase Intention</i> setelah Perbaikan Layanan .....	17
Gambar 3.2	Kategori Perbaikan terhadap Kegagalan Pelayanan Lelang <i>Online</i> .....	18
Gambar 3.3	Aspek Bisnis dalam <i>Software Engineering</i> .....	20
Gambar 3.4	<i>Essential Attributes of Good Software</i> .....	21
Gambar 3.5	Diagram Kasus Penggunaan Aplikasi .....	32
Gambar 3.6	Diagram Kasus Penggunaan Manajemen Autentikasi Pengguna.....	33
Gambar 3.7	Diagram Kasus Penggunaan Manajemen Traksaksi Lelang .....	37
Gambar 3.8	Diagram Kasus Penggunaan Manajemen Barang Lelang .....	42
Gambar 3.9	Diagram Kasus Penggunaan Manajemen Interaksi Antarpengguna .....	47
Gambar 3.10	Diagram Kasus Penggunaan Manajemen Kupon .....	52
Gambar 3.11	Arsitektur Dasar Aplikasi.....	57
Gambar 3.12	Arsitektur dan Teknologi Aplikasi.....	62
Gambar 3.13	Komponen Penyusun Struktur Aplikasi.....	63
Gambar 3.14	<i>Conceptual Database Model</i> (PDM) Aplikasi..	66
Gambar 3.15	<i>Physical Database Model</i> (PDM) Aplikasi .....	67
Gambar 4.1	<i>Error CORS</i> pada <i>Console Browser</i> .....	82
Gambar 4.2	<i>Whitelisting</i> Berhasil Dijalankan.....	88
Gambar 4.3	Detail Informasi <i>Email</i> pada Kotak Masuk Pengguna.....	88
Gambar 4.4	Halaman Antarmuka Registrasi .....	89
Gambar 4.5	Halaman Antarmuka <i>Login</i> .....	92
Gambar 4.6	Halaman Antarmuka Melihat Barang yang Dilelang .....	96
Gambar 4.7	Halaman Antarmuka Menawar Barang .....	97

Gambar 4.8	Halaman Antarmuka Mendaftarkan Barang untuk Dilelang .....	101
Gambar 4.9	Halaman Antarmuka Melihat Barang yang Pernah Didaftarkan .....	115
Gambar 4.10	Halaman Antarmuka Implementasi Menambahkan <i>Review</i> .....	117
Gambar 4.11	Halaman Antarmuka Implementasi Menambahkan <i>Review</i> .....	117
Gambar 4.12	Halaman Antarmuka Kasus Penggunaan Mengirimkan Pesan .....	121
Gambar 4.13	Halaman Antarmuka Melihat Kotak Pesan .....	127
Gambar 4.14	Halaman Antarmuka Memasukkan Kupon .....	132
Gambar 4.15	Halaman Antarmuka Kasus Penggunaan Melihat Daftar Pengguna .....	135
Gambar 4.16	Halaman Antarmuka Kasus Penggunaan Menambahkan Kupon .....	138
Gambar 4.17	Halaman Antarmuka Kasus Penggunaan Melihat Daftar Kupon .....	139
Gambar 5.1	Diagram Perbandingan Pengujian <i>User Experience</i> Pengguna .....	167
Gambar 5.2	Diagram Batang Hasil Pengujian Kecepatan Sistem .....	168
Gambar 5.3	Diagram Lingkaran Hasil Pengujian Kecepatan	169
Gambar 5.4	Diagram Evaluasi Pengujian <i>Maintainability</i> ...	170
Gambar 2.1	Kuisisioner Pengujian <i>User Experience</i> Responden 1 .....	189
Gambar 2.2	Kuisisioner Pengujian <i>User Experience</i> Responden 2 .....	190
Gambar 2.3	Kuisisioner Pengujian <i>User Experience</i> Responden 3 .....	191
Gambar 2.4	Kuisisioner Pengujian <i>User Experience</i> Responden 4 .....	192



Gambar 2.5	Kuisisioner	Pengujian	<i>User Experience</i>	
	Responden 5	.....		193
Gambar 2.6	Kuisisioner	Pengujian	<i>User Experience</i>	
	Responden 6	.....		194
Gambar 2.7	Kuisisioner	Pengujian	<i>User Experience</i>	
	Responden 7	.....		195
Gambar 2.8	Kuisisioner	Pengujian	<i>User Experience</i>	
	Responden 8	.....		196
Gambar 2.9	Kuisisioner	Pengujian	<i>User Experience</i>	
	Responden 9	.....		197
Gambar 2.10	Kuisisioner	Pengujian	<i>User Experience</i>	
	Responden 10	.....		198
Gambar 2.11	Kuisisioner	Pengujian	<i>User Experience</i>	
	Responden 11	.....		199

*(Halaman ini sengaja dikosongkan)*

## DAFTAR KODE SUMBER

IV.1	<i>Script package.json</i> Aplikasi . . . . .	85
IV.2	<i>Script</i> Webpack yang Belum Dioptimasi . . . . .	86
IV.3	<i>Script</i> Webpack yang Dioptimasi . . . . .	86
IV.4	<i>Including Script</i> pada <i>View</i> . . . . .	86
IV.5	Implementasi Antarmuka Registrasi . . . . .	90
IV.6	Implementasi Antarmuka <i>Login</i> . . . . .	92
IV.7	Implementasi <i>Back-end</i> Melihat Daftar Barang . . . . .	93
IV.8	Implementasi <i>Vue</i> Melihat Daftar Barang . . . . .	94
IV.9	Implementasi <i>Back-end</i> Menampilkan Halaman Lelang Barang . . . . .	97
IV.10	Implementasi Logika Lelang (menggunakan <i>Node.js</i> ) . . . . .	98
IV.11	Implementasi Logika UI (menggunakan <i>jQuery</i> ) . . . . .	99
IV.12	Implementasi <i>Back-end</i> Mendaftarkan Barang untuk Dilelang . . . . .	102
IV.13	Implementasi <i>Back-end</i> Mendaftarkan Barang untuk Dilelang . . . . .	103
IV.14	Implementasi <i>Back-end Upload</i> Gambar Barang . . . . .	103
IV.15	Implementasi <i>Back-end</i> Memperbarui Barang . . . . .	107
IV.16	Implementasi <i>View</i> Memperbarui Barang . . . . .	109
IV.17	Implementasi <i>Back-end Upload</i> Gambar Barang . . . . .	112
IV.18	Implementasi <i>Back-end</i> Melihat Barang yang Pernah Didaftarkan . . . . .	116
IV.19	Implementasi <i>Back-end</i> Menambahkan Review . . . . .	118
IV.20	Implementasi <i>Back-end</i> Mengirimkan Pesan . . . . .	122
IV.21	Implementasi Logika <i>View</i> Lelang (menggunakan <i>Node.js</i> ) . . . . .	122
IV.22	Implementasi Logika Pengiriman & Penerimaan Pesan (menggunakan <i>jQuery</i> ) . . . . .	124
IV.23	Implementasi <i>Back-end</i> Melihat Kotak Pesan . . . . .	128
IV.24	Implementasi Logika <i>View</i> Lihat Kotak Pesan (menggunakan <i>Node.js</i> ) . . . . .	129
IV.25	Implementasi <i>Back-end</i> Memasukkan Kupon . . . . .	133
IV.26	Implementasi <i>View Logic</i> Memasukkan Kupon . . . . .	133

IV.27	Implementasi Antarmuka Melihat Daftar Pengguna . . . . .	136
IV.28	Implementasi Antarmuka Menambahkan Kupon	137
IV.29	Implementasi Antarmuka Melihat Daftar Kupon	139
IV.30	Implementasi <i>Back-end</i> Kasus Penggunaan Memperbarui kupon . . . . .	140
A.1	Implementasi Node.js + Socket.io untuk Lelang	179
A.2	Implementasi Node.js + Socket.io untuk <i>Chat</i> . .	184

# BAB I

## PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar Tugas Akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan Tugas Akhir, dan sistematika penulisan.

### 1.1 Latar Belakang

Transaksi jual beli saat ini sudah dapat dilakukan lewat berbagai cara, antara lain menggunakan *e-commerce*, atau lewat *social media*, atau bisa dengan melelang di aplikasi lelang *online*. Sedikit berbeda dengan teknik penjualan di lelang *online*, karena aplikasi ini dapat diakses oleh banyak orang, tentu saja pelelang (*auctioneer*) tidak terbatas pada ruang lelang saja, tapi bisa berasal dari manapun selama mereka mengakses aplikasi tersebut. Lelang *online* ini tentu saja mendatangkan banyak manfaat, selain biaya yang lebih efisien dan hemat, dan juga tidak menguras waktu karena siapapun, kapanpun, dimanapun dapat mengajukan penawaran ataupun melelang barangnya tanpa harus pergi ke instansi tertentu dan melakukan lelang dengan cara konvensional.

Aplikasi serupa telah banyak, namun banyak aspek yang kurang dalam aplikasi tersebut, seperti informasi dari lelang tidak *reliable* (misal: stok barang ternyata sudah habis), alur proses yang tidak jelas sehingga membingungkan pengguna aplikasi, informasi yang kurang jelas, dan produk yang didapatkan ternyata tidak sesuai dengan informasi pada saat produk dilelang (*bad information*) [1]. Dan dari masalah teknis aplikasi, beberapa sumber menyatakan bahwa ketidakjelasan alur proses yang kurang diperhatikan oleh para developer aplikasi lelang *online* menjadi beberapa alasan yang kuat mengapa lelang *online* masih kurang diminati [2].

Diharapkan, dengan adanya aplikasi ini, beberapa kelemahan

yang masih ada pada aplikasi lelang *online* saat ini dapat diperbaiki, dan juga dapat membantu proses *online* yang ada di Indonesia, dan juga mampu memperbaiki citra aplikasi lelang *online* sehingga mampu meningkatkan minat masyarakat terhadap lelang *online*.

## 1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini adalah sebagai berikut:

1. Bagaimana cara membangun aplikasi lelang *online* berbasis web?
2. Bagaimana bentuk rancangan arsitektur aplikasi dan fitur yang sesuai dengan hasil analisa kelemahan aplikasi serupa dan strategi penyelesaian sesuai dengan paper acuan [1]?

## 1.3 Batasan Masalah

Dari permasalahan yang telah diuraikan di atas, terdapat beberapa batasan masalah pada tugas akhir ini, yaitu:

1. Aplikasi berbasis *web* dengan bahasa pemrograman PHP.
2. Aplikasi berbasis kerangka kerja Laravel.
3. Basis data yang digunakan adalah PostgreSQL.
4. Aplikasi tidak mencakup proses pembayaran.

## 1.4 Tujuan

Tujuan dari pengerjaan Tugas Akhir ini adalah:

1. Membangun aplikasi lelang *online* berbasis *web* yang lebih kredibel sesuai dengan paper yang dijadikan acuan pada tugas akhir ini.

## 1.5 Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan Tugas Akhir ini yaitu:

### 1. Studi Literatur & Observasi

Pada tahap ini dilakukan pengumpulan dan penggalan informasi lewat literatur maupun artikel-artikel dari internet, yang diperlukan dalam proses perancangan dan implementasi sistem.

### 2. Analisa dan Perancangan Sistem

Pada tahap ini, dilakukan analisa dan pendefinisian kebutuhan sistem yang digunakan untuk masalah yang dihadapi. Tahapan-tahapannya adalah sebagai berikut:

- a analisa aktor yang terlibat dalam sistem;
- b perancangan model kasus penggunaan;
- c perancangan bisnis proses dalam aplikasi;
- d analisa masalah-masalah yang sering muncul saat penulis membuat aplikasi sebelumnya;
- e perancangan dan desain arsitektur aplikasi; dan
- f perancangan antarmuka aplikasi.

### 3. Implementasi

Tahap ini merupakan implementasi dari rancangan yang telah dibuat pada tahap sebelumnya.

### 4. Pengujian dan Evaluasi

Pada tahap ini, dilakukan pengujian terhadap aplikasi terhadap fungsionalitas dan non-fungsionalitas aplikasi.

## 1.6 Sistematika Penulisan

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini:

### Bab I Pendahuluan

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan Tugas Akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan Tugas Akhir.

## **Bab II Dasar Teori**

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan mendasari pembuatan Tugas Akhir ini.

## **Bab III Analisa dan Perancangan Sistem**

Bab ini membahas mengenai analisa dan perancangan aplikasi. Perancangan aplikasi meliputi perancangan data, arsitektur, proses dan struktur program.

## **Bab IV Implementasi**

Bab ini berisi deskripsi lengkap implementasi aplikasi.

## **Bab V Pengujian dan Evaluasi**

Bab ini membahas pengujian untuk menguji apakah aplikasi sudah tepat sasaran pada kebutuhan fungsional dan non-fungsional yang dirumuskan pada tahap Analisa dan Perancangan Sistem (Bab III).

## **Bab VI Kesimpulan dan Saran**

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan, dan membahas saran beserta *further enhancements* untuk pengembangan sistem lebih lanjut.

### **Daftar Pustaka**

Merupakan daftar referensi yang digunakan untuk mengembangkan Tugas Akhir.

### **Lampiran**

Merupakan bab tambahan yang berisi hal-hal terkait yang penting dalam aplikasi ini.



## BAB II

### LANDASAN TEORI

#### 2.1 Amazon Web Service

*Amazon Web Services* adalah sekumpulan layanan-layanan berbasis Cloud Computing yang di sediakan oleh Amazon sejak tahun 2002. Meskipun salah satu perusahaan raksasa internet ini sering kita kenal untuk membeli buku dan lagu, namun sekarang Amazon telah menambah layanannya dalam hal infrastruktur *cloud computing*. *Amazon Web Services* ini menyediakan layanan-layanan nya yang saling terintegrasi dan mudah kustomisasi. Pada tahun 2006, amazon mengenalkan Amazon's *Elastic Compute Cloud* (EC2) sebagai *commercial web service* yang menyediakan akses *cloud* kepada perusahaan dan individu untuk menyewa komputer *storage* yang bisa digunakan sebagai *platform* pengembangan aplikasi secara *online*, inilah awal dari IaaS, yaitu perusahaan yang menyediakan infrastruktur sebagai sebuah layanan[3].

Dalam website resminya, disebutkan bahwa AWS dapat membantu aplikasi menjadi lebih cepat, lebih aman, dan menghemat *costs* dengan *scaling performance* menggunakan teknologi *cloud computing*[4].

#### 2.2 Concurrency

Konkurensi adalah bisa dikatakan sebagai suatu fitur di mana *database management system* (DBMS) mengijinkan banyak transaksi pada saat bersamaan untuk mengakses data yang sama. Dalam melakukan konkurensi, dibutuhkan suatu *Concurrency Control Mechanism* (CCM) agar transaksi yang dilakukan oleh banyak user pada suatu sistem di dalam waktu yang bersamaan tidak saling “mengganggu” dan tidak menghasilkan *inconsistency* data.

Tiga masalah umum yang muncul dalam konkurensi adalah

sebagai berikut:

1. *Lost Update Problem*

Masalah operasi update yang sukses dari seorang pengguna kemudian ditimpali oleh operasi update dari pengguna lain.

2. *Uncomited dependency problem* (ketergantungan yg tidak sukses/modifikasi sementara)

3. *Inconsistent analysis problem* [5]

### 2.3 *Data Growth*

Yang dimaksud dalam *data growth* pada *section* ini adalah seberapa cepat perkembangan jumlah data yang disimpan oleh server. Data tersebut bisa berupa *row* dalam *database* ataupun data gambar, video dll[6].

Sejak tahun 2000an, perkembangan data meningkat pesat dan memunculkan bisnis penyedia *data storage* dan *networking equipment*.

Sebagai ilustrasi, pada buku referensi tercatat bahwa *data equilibrium flow* firma-firma *e-Commerce* pada dekade 1990-2000 meningkat hingga 1.5 *billion gigabytes* setiap tahunnya[7].

### 2.4 *JSON Web Token*

*JSON Web Token* atau lebih dikenal dengan JWT, merupakan sebuah *token* berbentuk JSON yang padat-informasi (ukurannya), informasi mandiri untuk ditransmisikan antar pihak yang terkait. *Token* tersebut ini dapat diverifikasi dan dipercaya karena sudah di-*sign* secara digital. *Token* JWT bisa di-*sign* dengan menggunakan *secret* (algoritma HMAC) atau pasangan *public / private key* (algoritma RSA) [8].

## 2.5 Laravel

Laravel adalah *framework* PHP yang dikembangkan pertama kali oleh Taylor Otwell. Walaupun termasuk baru, namun komunitas pengguna laravel sudah berkembang pesat dan mampu menjadi alternatif utama dari sejumlah *framework* besar seperti CodeIgniter dan Yii. Laravel oleh para *developer* disetarakan dengan CodeIgniter dan FuelPHP namun memiliki keunikan tersendiri dari sisi *coding*. Laravel memiliki beberapa keunggulan, diantaranya:

1. Sintaks yang sederhana dan *programmer-friendly*
2. Tersedia *generator* yang canggih dan memudahkan, Artisan CLI
3. Fitur *Schema Builder* untuk berbagai *database*
4. Fitur *Migration* dan *Seeding* untuk berbagai *database*
5. Fitur *Query Builder* yang powerful
6. Eloquent ORM (*Object Relational Mapping*)
7. Fitur pembuatan *package* dan *bundle*
8. *Dependency Injection* [9]

## 2.6 Laravel Dusk

Laravel Dusk adalah sebuah fitur baru yang ditujukan untuk *functional testing*, yang baru diluncurkan dan dinamakan secara *default* pada Laravel versi 5.4. Dalam *site* dokumentasinya, disebutkan bahwa Laravel Dusk menyediakan *browser automation & testing API* yang ekspresif dan mudah digunakan. Secara otomatis, Dusk tidak memerlukan instalasi JDK atau Selenium pada *host*, namun menggunakan *ChromeDriver standalone*, namun juga dapat menggunakan driver Selenium yang kompatibel [10].

## 2.7 Lelang

Lelang adalah proses membeli dan menjual barang atau jasa dengan cara menawarkan kepada penawar, menawarkan tawaran harga lebih tinggi, dan kemudian menjual barang kepada penawar harga tertinggi. Dalam teori ekonomi, lelang mengacu pada beberapa mekanisme atau peraturan perdagangan dari pasar modal [11].

Lelang menurut sejarahnya berasal dari bahasa Latin *auctio* yang berarti peningkatan harga secara bertahap. Para ahli menemukan bahwa dalam literatur Yunani, lelang telah dikenal 450 tahun sebelum Masehi. Jenis lelang yang populer saat itu antara lain adalah karya seni, tembakau, kuda, budak dan sebagainya[12].

## 2.8 Lelang Daring/Lelang *Online*

Lelang adalah proses membeli dan menjual barang atau jasa dengan cara menawarkan kepada penawar, menawarkan tawaran harga lebih tinggi, dan kemudian menjual barang kepada penawar harga tertinggi. Dalam teori ekonomi, lelang mengacu pada beberapa mekanisme atau peraturan perdagangan dari pasar modal.

Sementara lelang daring atau lelang melalui internet muncul seiring dengan perkembangan internet. Barang atau jasa yang diperjualbelikan dipasang di situs dan peserta lelang dapat mengikuti acara lelang secara daring. Perusahaan lelang yang berhasil menggunakan sarana internet salah satunya adalah *Ebay*. Di Indonesia, lelang melalui internet sudah dipelopori oleh pemerintah dengan situs lelang *online* yang dapat diakses melalui website resmi Kemenkeu [13]. Berikut adalah beberapa istilah yang ada dalam lelang online:

1. BID atau *Bidding*, artinya: Menawarkan

2. BIN (*Buy In Now*) artinya: Beli sesuai harga yang telah ditawarkan penjual
3. INC (*Increment*) artinya: Minimum kenaikan *bid* setelah *bid* sebelum nya [14]

## 2.9 MongoDB

MongoDB (dari *humongous*) adalah sistem basis data berorientasi dokumen lintas *platform*. Diklasifikasikan sebagai basis data "NoSQL", MongoDB menghindari struktur basis data relasional tabel berbasis tradisional yang mendukung JSON seperti dokumen dengan skema dinamis (MongoDB menyebutnya sebagai format BSON), membuat integrasi data dalam beberapa jenis aplikasi lebih mudah dan lebih cepat. Dirilis di bawah kombinasi dari GNU *Affero General Public License* dan Lisensi Apache, MongoDB adalah perangkat lunak bebas dan sumber terbuka[15].

## 2.10 Node.js

Node.js adalah *platform server-side* perangkat lunak dan aplikasi jaringan. Ditulis dengan bahasa javascript dan bisa dijalankan pada Windows, Mac OS X dan Linux tanpa perubahan kode program. Node.js memiliki pustaka server HTTP sendiri sehingga memungkinkan untuk menjalankan *webserver* tanpa menggunakan program *webserver* seperti Apache atau Lighttpd [16].

## 2.11 NoSQL

NoSQL adalah istilah yang dikenal dalam teknologi komputasi untuk merujuk kepada kelas yang luas dari sistem manajemen basis data yang diidentifikasi dengan tidak

mematuhi aturan pada model sistem manajemen basis data relasional yang banyak digunakan.

NoSQL tidak dibangun terutama dengan tabel dan umumnya tidak menggunakan SQL untuk memanipulasi data, sehingga sering ditafsirkan sebagai “tidak hanya SQL” [17].

## 2.12 npm / *Node Package Manager*

NPM memiliki dua fungsi utama, yaitu sebagai repositori *online* yang berisi banyak *package* atau modul untuk aplikasi NodeJS dan yang kedua adalah sebuah utilitas baris perintah (*command line*) yang digunakan untuk menginstal paket-paket yang dibutuhkan dan juga untuk mengelola versi dan *package dependencies* dari NodeJS. Dengan NPM Anda akan mudah mencari, pemasangan/*installation*, *uninstall* aplikasi atau *module/package* Node.js[18].

## 2.13 PostgreSQL

PostgreSQL adalah sebuah produk *database* relasional yang termasuk dalam kategori *free open source software (FOSS)*. PostgreSQL terkenal karena fitur-fitur yang *advance* dan pendekatan rancangan modelnya menggunakan paradigma *object-oriented*, sehingga sering dikategorikan sebagai *Object Relational Database Management System (ORDBMS)*. Beberapa fitur PostgreSQL adalah sebagai berikut:

1. *Inheritance*, dimana satu table dapat diturunkan model dan beberapa karakteristik dari table lainnya.
2. *Multi-Version Concurrency Control* dimana user diberi data snapshot ketika suatu perubahan dilakukan sampai perubahan tersebut *dicommit*.
3. *Rules*, dimana suatu *query* DML yang dikirimkan ke server akan mengalami penulisan ulang (*rewrite*). Ini terjadi

- sebelum diproses oleh *query planner*.  
 4. dan berbagai fitur lainnya [19].

## 2.14 Repository Pattern

*Repository Pattern* adalah sebuah pola dalam struktur *software engineering* yang memisahkan *data management layer* ke dalam sebuah layer tersendiri - yang dihandle oleh sebuah bagian struktur yang disebut repository.

Jika menggunakan *pattern* ini, semua kode spesifik yang terkait dengan *persistence logic* dan implementasi akses data berhenti sampai di *repository* (*controller* hanya *redirecting request* dan validasi *request*)[20].

Dalam sebuah referensi, disebutkan bahwa: "*The repository pattern covers large centralized transaction-oriented databases, the blackboard systems used for some AI applications, and systems with predetermined execution patterns in which different phases add information to a single complex data structures (e.g., compilers). These variants differ chiefly in their control structure.*" [21].

## 2.15 SendGrid

SendGrid adalah sebuah *customer communication platform* untuk *email marketing* dan transaksional yang berbasis di Denver, Colorado.

SendGrid menyediakan layanan pengiriman *email* yang berbasis *cloud* kepada pihak bisnis. Layanan yang ditawarkan sangat beragam, mulai dari *shipping notifications*, *friend requests* dan lain lagi.

Selain itu, juga dapat *handling ISP monitoring*, *domain keys*, *feedback loops*, dan juga memberikan report *opened mails*, *unsubscribes*, *bounces* dan *spam reports*. Pada tahun 2012,

SendGrid menggaet Twilio dan menambahkan layanan integrasi SMS, Suara dan *push notification*[22].

### 2.16 *Service Worker*

*Service worker* adalah *script* yang dijalankan browser Anda di latar belakang, terpisah dari laman web, yang membuka pintu ke berbagai fitur yang tidak memerlukan laman *web* atau interaksi pengguna. Saat ini, *service worker* sudah menyertakan berbagai fitur seperti pemberitahuan push dan sinkronisasi latar belakang. Di masa mendatang, *service worker* akan mendukung hal-hal lainnya seperti sinkronisasi berkala atau *geofencing*. Fitur inti yang didiskusikan dalam tutorial adalah kemampuan mencegat dan menangani permintaan jaringan, termasuk mengelola *cache* respons lewat program[23].

### 2.17 *SMTP / Simple Mail Transfer Protocol*

SMTP adalah suatu protokol yang digunakan untuk mengirimkan pesan elektronik (*e-mail*) antar-*server*, yang bisa dianalogikan sebagai kantor pos. Ketika kita mengirim sebuah *e-mail*, komputer kita akan mengarahkan *e-mail* tersebut ke sebuah SMTP *server*, untuk diteruskan ke *mail server* tujuan [24].

### 2.18 *Socket.io*

Socket.io adalah *library* Javascript untuk aplikasi *web* yang bersifat *realtime*. Socket.io menjembatani antara komunikasi dua arah antara *web clients* dan *server*. Socket.io terbagi menjadi dua bagian, yaitu *client-side library* yang berjalan di browser client, dan *server-side library* yang menggunakan Node.js. Kedua komponen tersebut mempunyai API yang sama. Seperti Node.js,



Socket.io juga bersifat *event-driven*. Socket.IO menggunakan protokol *websocket* dengan *polling* sebagai opsi *fallback*. Meskipun Socket.IO merupakan ‘pembungkus’ untuk soket *web*, namun ia memiliki banyak fitur, antara lain *broadcast* ke banyak soket, dan I/O yang asinkronus [25].

### 2.19 Test Script

Test Script dalam dunia *software testing* adalah set instruksi atau sekumpulan baris kode yang akan melakukan *testing* terhadap fungsionalitas sistem dengan target tertentu [26].

Ada beberapa jenis *script test*:

1. *Manual testing*, atau lebih sering disebut *test cases*
2. *Automated Testing*

### 2.20 Vendu Reglement

Lelang dilegalisasi & resmi masuk Indonesia dalam perundang-undangan sejak 1908, yaitu dengan berlakunya *Vendu Reglement*, Stbl. 1908 No. 189 dan *Vendu Instructie*, Stbl 1908 No. 190. *Vendu Reglement* ini berisikan peraturan-peraturan dasar lelang yang berlaku hingga saat ini, dan menjadi dasar hukum penyelenggaraan lelang di Indonesia[11].

### 2.21 Vue.js / Vue

Vue adalah sebuah *framework* Javascript yang *progressive* dan bersifat *open-source* untuk membangun *user interface*. Integrasi kedalam project yang menggunakan *library* Javascript lain menjadi lebih mudah dengan Vue karena Vue memang didesain untuk *incrementally adoptable*. Vue juga dapat berfungsi sebagai *web application framework* untuk membangun *single-page applications*.

Dalam 2016 *Javascript Survey*, Vue mendapatkan 89% untuk kategori *developer satisfaction rating*. Vue mengakumulasikan 98 bintang Github setiap hari, dan menduduki peringkat ke-10 *project* Github dengan bintang terbanyak sepanjang waktu[27].

## 2.22 *Whitelist*

*Whitelist* adalah memindahkan daftar alamat atau domain dari pengirim *email* dengan harapan agar muncul pada kotak pesan *email* utama Anda. Sederhananya, hal ini dilakukan untuk menghindari agar pesan *email* yang dikirimkan oleh pengirim pesan sebenarnya tidak terbaca sebagai *spam*[22].

## **BAB III**

### **ANALISA DAN PERANCANGAN SISTEM**

#### **3.1 Analisa Sistem**

Pada subbab ini, penulis akan memaparkan analisa terhadap kebutuhan baik fungsional dan non-fungsional, teknis dan serta pengaruhnya dalam perancangan aplikasi pada Subbab 3.2. Subbab Analisa Sistem akan dipaparkan secara sistematis sebagai berikut:

- a. analisa kebutuhan dasar;
- b. analisa aspek bisnis;
- c. analisa kebutuhan fungsional;
- d. analisa kebutuhan non-fungsional; dan
- e. rancangan arsitektur, struktur dan teknologi yang digunakan.

##### **3.1.1 Deskripsi Umum Aplikasi**

*Bid! Bid! Bid!* Istilah tersebut pasti tidak asing bagi para pecinta lelang online. Lelang sendiri adalah proses membeli dan menjual barang atau jasa dengan cara menawarkan kepada penawar, menawarkan tawaran harga lebih tinggi, dan kemudian menjual barang kepada penawar harga tertinggi [28]. Di Indonesia, sistem lelang sudah digunakan sejak jaman Hindia Belanda dimana saat itu sistem lelang pertama kali diperkenalkan di Indonesia dan biasa digunakan lelang terhadap aset-aset pejabat atau pemerintah yang dimutasi pada saat itu.

Dalam proses bisnisnya sendiri, lelang cukup sederhana, yaitu pengguna dapat melelang dan dapat menjual barang untuk dilelang. Pengguna dapat manajemen riwayat lelang dan manajemen barang yang ia daftarkan untuk dilelang. Hal ini akan didefinisikan lebih lanjut dalam Subbab 3.1.5.7, yaitu pada bagian Spesifikasi Kebutuhan Fungsional.

### 3.1.2 Analisa Paper Rujukan

Bercermin terhadap aplikasi *e-commerce* yang telah ada, masalah yang paling sering dialami adalah ketidakpuasan pengguna. Salah satu indikator bahwa suatu perusahaan dikatakan memiliki ketidakpuasan pelanggan adalah karena kegagalan dalam pelayanannya. Seorang pelanggan sangat mungkin memutuskan untuk komplain setelah mengalami ketidakpuasan terhadap layanan suatu perusahaan, dan jika tidak ditangani dengan baik, hal ini bisa berakibat fatal terhadap reputasi dan kepercayaan pengguna terhadap aplikasi tersebut.

Oleh karena itu, sebuah paper mengangkat topik ini khusus dalam bidang aplikasi lelang online, menganalisa kegagalan dan ketidakpuasan pengguna, beserta solusi-solusi yang ditawarkan oleh pengguna aplikasi untuk memperbaiki kegagalan pelayanan tersebut.

Pada Gambar 3.1, dijabarkan beberapa jenis kegagalan yang pernah dialami oleh pengguna aplikasi serta fatalitas/pengaruh buruk kegagalan tersebut terhadap kepercayaan pengguna. Pada Gambar 3.2 dipaparkan tingkat kepuasan pengguna terhadap solusi *recovery* yang diberikan.

Berdasarkan hasil analisa tersebut dan mencocokkan dengan batasan Tugas Akhir ini, fitur-fitur yang ditambahkan selain daripada fitur dasar aplikasi lelang *online* sebagai *added value* adalah sebagai berikut:

1. Fitur *chatting*, untuk mengurangi kemungkinan *Bad Information* dimana ekspektasi dan persepsi terhadap barang yang dilelang antara pembeli dan penjual tidak sama dan *Special Needs*; dan
2. Fitur pemberian kupon voucher (*Discount and Correction Plus*) yang bisa berupa *free shipping* atau *discount*.

Type of service failure	Severity of service failures	Satisfaction with recovery	Repeat purchase intention with recovery
<i>Group 1 service delivery system failures</i>			
Packaging problem	7.1 <sup>a</sup> (2.5) <sup>b</sup>	6.3 (3.1)	5.9 (3.1)
Slow/unavailable service	6.9 (2.5)	4.9 (3.1)	4.4 (3.2)
Product defect	7.5 (2.6)	5.5 (3.4)	4.9 (3.4)
Out of stock	7.2 (2.3)	5.3 (2.9)	5.1 (3.3)
Bad information	7.7 (2.4)	4.2 (3.2)	3.6 (3.1)
Alterations and repairs	8.1 (2.7)	2.2 (2.2)	2.0 (1.9)
Hold disaster	7.8 (1.8)	2.6 (1.7)	3.0 (2.1)
Pricing failure	7.8 (3.0)	5.2 (3.6)	5.7 (3.6)
Policy failure	7.1 (2.1)	3.7 (2.9)	3.2 (2.8)
Subtotal, Group 1	7.2 (2.5)	5.3 (3.2)	4.9 (3.3)
<i>Group 2 Buyer needs and requests</i>			
Gap between expectation and perception	7.9 (2.2)	2.7 (2.6)	2.4 (2.5)
Size variation	6.7 (2.0)	5.9 (3.6)	5.9 (3.6)
Special order or request	8.6 (1.6)	3.5 (3.5)	3.3 (3.1)
Admitted buyer error	4.0 (1.7)	5.7 (2.1)	4.0 (3.5)
Subtotal, Group 2	7.7 (2.2)	3.3 (3.0)	3.0 (3.0)
<i>Group 3 Unprompted and unsolicited seller actions</i>			
Seller attention failures	7.6 (2.4)	3.5 (2.8)	3.3 (2.8)
Seller-created embarrassments	7.7 (2.8)	3.5 (2.7)	3.8 (3.1)
Seller fraud problem	9.4 (1.2)	1.1 (0.4)	1.0 (0.0)
Mischarged	6.8 (2.9)	5.4 (4.0)	5.2 (3.9)
Leak of personal data	9.5 (0.7)	1.0 (0.0)	1.0 (0.0)
Subtotal, Group 3	8.0 (2.4)	3.1 (2.8)	3.0 (2.9)
Total	7.4 (2.5)	4.8 (3.3)	4.4 (3.3)

<sup>a</sup> Mean.

<sup>b</sup> Standard deviation.

**Gambar 3.1** Fatalitas Kegagalan dalam Aplikasi Lelang *Online* dan Kepuasan Perbaikan Pelayanan dan *Repeat Purchase Intention* setelah Perbaikan Layanan

Recovery strategy	No. (%)	No. of satisfactory recovery	No. of dissatisfactory recovery	Satisfaction with recovery	Retention with recovery
Correction	280 <sup>a</sup> (32.3) <sup>b</sup>	228 <sup>a</sup> (81.4) <sup>b</sup>	52 <sup>a</sup> (18.6) <sup>b</sup>	7.2 <sup>c</sup> (2.4) <sup>d</sup>	6.6 <sup>c</sup> (2.7) <sup>d</sup>
Correction plus	82 (9.5)	77 (93.9)	5 (6.1)	7.7 (2.1)	7.6 (2.5)
Discount	26 (3.0)	23 (88.5)	3 (11.5)	7.0 (2.2)	6.0 (2.9)
Replacement	14 (1.6)	2 (14.3)	12 (85.7)	3.4 (2.7)	3.5 (2.7)
Store credit	16 (1.8)	9 (56.3)	7 (43.7)	6.1 (2.9)	6.4 (2.7)
Apology	54 (6.2)	20 (37.0)	34 (63.0)	3.7 (2.6)	3.4 (2.6)
Refund	67 (7.7)	42 (62.7)	25 (37.3)	6.4 (2.9)	5.2 (3.1)
Unsatisfactory correction	182 (21.0)	0 (0)	182 (100.0)	1.9 (1.3)	1.8 (1.6)
Failure	39 (4.5)	0 (0)	39 (100)	1.3 (0.9)	1.3 (0.9)
Nothing	107 (12.4)	0 (0)	107 (100)	1.7 (1.2)	1.7 (1.6)
Total	867 (100)	407 (46.9)	460 (53.1)	4.8 (3.3)	4.4 (3.3)

<sup>a</sup> Number.

<sup>b</sup> %.

<sup>c</sup> Mean.

<sup>d</sup> Standard deviation.

**Gambar 3.2** Kategori Perbaikan terhadap Kegagalan Pelayanan Lelang *Online*

### 3.1.3 *Bussiness Aspects of Software Engineering*

Lelang merupakan salah satu metode pertukaran barang dan jasa dengan metode penetapan harga yang berbeda dengan perdagangan. Oleh karena itu, lelang juga termasuk dalam kategori bisnis. Yang menarik adalah, ketika bisnis digabungkan dengan teknologi atau yang sering disebut *e-commerce*, hal yang sekedar pertukaran barang bertransformasi menjadi sebuah

sistem interaktif yang kompleks dimana tujuan utamanya adalah menarik pengunjung/pengguna untuk menyelesaikan sebuah transaksi. Hal ini tentu sangat krusial, penting, dan tertantang untuk menyelesaikannya.

Dalam mencapai kesuksesan dan tingkat kompetitif yang tinggi, haruslah menyediakan layanan dengan kesan *user experience (UX)* yang positif bagi para penggunanya. Morville [29, p. 27] , dalam studi yang dilakukannya, menyebutkan bahwa UX tercakup dalam 6 aspek esensial, yaitu

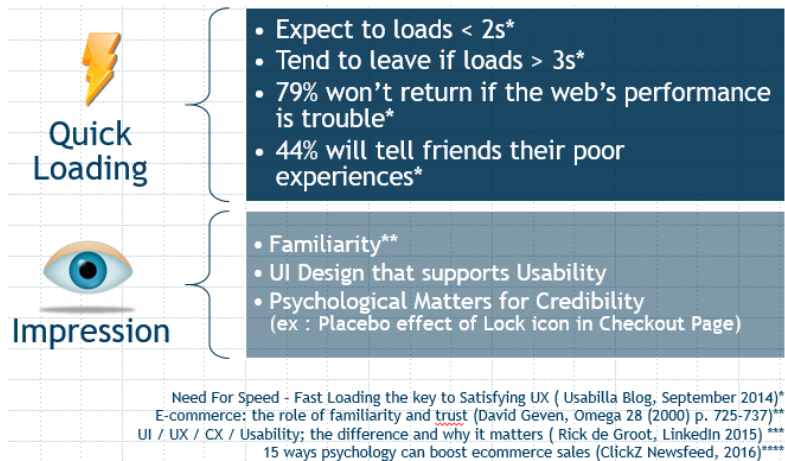
- a. *useful*
- b. *usable*
- c. *findable*
- d. *desirable*
- e. *accessible*
- f. *credible*.

Hasil-hasil temuan penting yang menarik dalam pengaruh *user experience*, adalah sebagai berikut dikutip dari sebuah sumber adalah sebagai berikut:

- a. *User tend to leave if a page loads more than 3 seconds;*
- b. *79% of users won't return if the web's performance and experience is poor; and*
- c. *44% of users will tell the poor experiences to their friends.*

Selain dari faktor *user experience* dan *performance*, beberapa hal yang menjadi poin penting dan menarik dalam beberapa studi yang terkait adalah sebagai berikut:

- a *Familiarity* - yang dapat didefinisikan sebagai tingkat familier atau kesamaan dengan sistem sejenis ternyata dapat membangun *trust* sehingga mensugesti pengguna untuk menyelesaikan transaksi yang dilakukan;
- b *Usability* yang memudahkan pengguna dalam menyelesaikan transaksi; dan
- c Aspek-aspek psikologi seperti pemilihan warna, penggunaan *icon* yang sesuai, seperti *icon* gembok pada



**Gambar 3.3** Aspek Bisnis dalam *Software Engineering*

halaman pembayaran ternyata dapat mengesankan *security* pada pengguna.

Dari hasil temuan ini, dapat disimpulkan bahwa *user experiences*, *performances*, *usability* dan psikologi memiliki pengaruh besar dalam kesuksesan lelang *online* dalam menarik hati para penggunanya. Hal ini akan mempengaruhi definisi kebutuhan fungsionalitas yang akan dibahas dalam Subbab 3.1.5.7.

### 3.1.4 *Technical Analysis*

Untuk membuat sebuah aplikasi yang sukses, tentunya banyak sekali aspek yang harus diperhatikan. Selain kualitas aplikasi yang akan dibuat, juga ketahanannya terhadap perubahan karena *e-commerce* adalah sesuatu yang sangat cepat berubah karena kompetitor yang sangat kompetitif dan dorongan teknologi yang membuat efektifitas dan efisiensi menjadi lebih baik.



Product characteristics	Description
Maintainability	Software should be written in such a way so that it can evolve to meet the changing needs of customers. This is a critical attribute because software change is an inevitable requirement of a changing business environment.
Dependability and security	Software dependability includes a range of characteristics including reliability, security, and safety. Dependable software should not cause physical or economic damage in the event of system failure. Malicious users should not be able to access or damage the system.
Efficiency	Software should not make wasteful use of system resources such as memory and processor cycles. Efficiency therefore includes responsiveness, processing time, memory utilization, etc.
Acceptability	Software must be acceptable to the type of users for which it is designed. This means that it must be understandable, usable, and compatible with other systems that they use.

**Gambar 3.4** *Essential Attributes of Good Software*

Dari aspek *software engineering* sendiri, *software engineering* dimaksudkan untuk menunjang/support pengembangan *software* daripada *individual programming*. Hal ini mencakup: **a) evolution** **b) design** **c) supporting program specification** [30].

Berdasarkan kriteria tersebut, maka setiap poin perlu diperhatikan agar dapat mengembangkan sebuah aplikasi yang tidak hanya sukses, tapi juga bertahan dalam kompetisi. Dalam istilah bisnis, hal ini disebut dengan *risk management & planning*.

### 3.1.5 Analisa Penulis

#### 3.1.5.1 Analisa User Experience dari E-Commerce di Indonesia

Selama masa pengerjaan aplikasi, penulis sering menganalisa dan memperhatikan kebiasaan-kebiasaan yang umum di website *e-commerce* di Indonesia. Salah satu yang paling sering dianalisa oleh penulis adalah adalah situs Tokopedia. Dalam pengembangannya, *user interface* aplikasi akan dipengaruhi

analisa ini, yang dijabarkan seperti berikut:

1. Halaman yang muncul bukanlah *eagerloading*, tapi *lazy loading*, yang merupakan solusi cerdas untuk mengakali *delay loading item* yang sudah pasti jumlahnya sangat banyak (maka butuh *query* yang tentunya memakan waktu cukup lama), namun juga memainkan faktor psikologi / *user behaviour* pengguna dengan membiarkan pengguna melihat tahap demi tahap halaman 'diisi'; dan
2. *User Interface* yang sederhana dan pemilihan warna yang *soft*.

### 3.1.5.2 Analisa Keamanan pada koneksi Soket

Untuk mengakomodasi fitur yang bersifat *realtime*, dibutuhkan koneksi ke soket secara terus menerus. Hal ini tentu dapat menjadi sasaran empuk *security* karena jika tidak diamankan, maka dapat menjadi peluang besar bagi para pihak yang tidak berkepentingan untuk merusak proses bisnis aplikasi.

Namun, jika dalam setiap koneksi soket harus mengirimkan *credentials*, hal ini tentu menjadi tidak praktis dan malah lebih berbahaya karena membiarkan data-data sensitif seperti *password* dan *username* berlalu-lalang di jaringan internet. Selain itu, *disadvantages*nya adalah ketidakpraktisan untuk selalu *mengquery* database setiap kali ada koneksi, tentu saja ini memperlambat kerja *database* dan menambah waktu *delay*. Maka dari itu, penulis mengidentifikasi poin-poin penting berikut :

- Hindari *query* database untuk autentikasi yang sifatnya masif
- Menggunakan mekanisme autentikasi yang menggunakan *credentials* karena rentan dengan masalah keamanan
- Mencari metode yang lebih efektif

### 3.1.5.3 Analisa *Best Practice* dalam Struktur Perangkat Lunak

Pada dasarnya, Laravel adalah kerangka kerja MVC. Namun, ada banyak fitur yang ada dalam aplikasi Lelang *online* ini yang tidak terakomodasi dalam MVC, misal sebagai berikut :

1. Sistem Verifikasi lewat *email* - yang berarti aplikasi harus berinteraksi dengan SMTP server
2. Sistem *Generate Token* JWT.io, dimana dalam proses *Generate Token* sama sekali tidak ada database dilibatkan. Jika fitur-fitur tersebut 'dipaksa' dimuat ke dalam MVC, maka tentu saja strukturnya menjadi ganjil, dan muncul *code smell* berikut :

1. *Large Class*, dimana terdapat satu buah file yang sangat panjang (biasanya merupakan entitas utama, dalam hal ini contohnya barang/item)
2. *Inappropriate Intimacy*, dimana terdapat satu kelas yang menyimpan *logic* yang tidak seharusnya ia simpan
3. *Duplicated Code*

Dari hasil analisa ini, penulis mengidentifikasi strategi-strategi yang akan diterapkan dalam rancangan struktur aplikasi pada Subbab 3.2.5, yaitu sebagai berikut:

1. Penggunaan Repository Pattern  
Memisahkan antara Data Processing Layer dan View Layer - agar lebih rapi, terstruktur, hal ini juga dapat menghindari *Duplicated Code*.
2. Penambahan Komponen : Service dan Provider  
Untuk memisahkan *logic* aplikasi yang terkait dengan akses *eksternal services*. Tujuannya, agar jika kedepannya terdapat perbaikan fitur/penambahan fitur, lebih mudah melakukan *traceback* terhadap file/kelas yang bertanggungjawab terhadap fitur tersebut.

#### 3.1.5.4 Analisa Aplikasi Serupa

Selama pengerjaan aplikasi, penulis menganalisa aplikasi serupa. Penulis menemukan aplikasi yang kurang lebih alur bisnis/alur penggunaan aplikasinya serupa yaitu : Carousell. Penulis melihat beberapa kesamaan antara sifat transaksi aplikasi tugas akhir saya dengan aplikasi tersebut, yaitu:

1. Sama-sama tidak mengakomodasi pembayaran
2. Sama-sama tidak adanya kepastian harga (bedanya, pada Carousell yang terjadi adalah *bargaining*

Sehingga dalam alur proses nya, banyak diadaptasi dari Carousell, agar pengguna dapat lebih familiar dan *predictability*nya lebih tinggi jika diadaptasi dari *c-commerce* lainnya yang lebih umum digunakan oleh pengguna.

#### 3.1.5.5 Analisa Aplikasi Serupa

Selama pengerjaan aplikasi, penulis menganalisa aplikasi serupa. Penulis menemukan aplikasi yang kurang lebih alur bisnis / alur penggunaan aplikasinya serupa yaitu : Carousell.

Penulis melihat ada beberapa kesamaan antara sifat transaksi aplikasi tugas akhir saya dengan aplikasi tersebut, yaitu :

1. Sama-sama tidak mengakomodasi pembayaran
2. Sama-sama tidak adanya kepastian harga (bedanya, pada Carousell yang terjadi adalah *bargaining*

Sehingga dalam alur proses nya, banyak diadaptasi dari Carousell, agar pengguna dapat lebih familiar dan *predictability*nya lebih tinggi jika diadaptasi dari *E-commerce* lainnya yang lebih umum digunakan oleh pengguna.

#### 3.1.5.6 Analisa Penyimpanan Data

Untuk penyimpanan data, terdapat 2 jenis data yang sifatnya cukup berbeda, yaitu sebagai berikut:

### 1. **Data transaksional disimpan di DBMS SQL - Relational**

Data yang sifatnya transaksional, seperti data *bidding*, data pengguna, dan lain sebagainya. Untuk data ini, lebih baik jika menggunakan database Postgre, untuk menjaga integritas data dan *integrity checking* juga menjadi lebih baik.

### 2. **Data non-transaksional disimpan di DBMS NoSQL**

Data *chatting*, data *joined rooms* kurang tepat jika disimpan dalam database transaksional karena sifat pertambahan datanya yang sangat cepat, masif dan urgensi integritas data tidak terlalu diprioritaskan (dibanding dengan data transaksional pada poin sebelumnya). Oleh karena itu, baiknya data ini disimpan pada database NoSQL dengan alasan-alasan sebagai berikut.

- Banyaknya transaksi *read write*;
- Ketidaksamaan frekuensi *read and write* data semua pengguna;
- Sifat permintaan transaksi yang cepat; dan
- Kemungkinan perubahan struktur atribut pada pesan (misal: *attachments, forwarding, replying*, dll) akan sangat menyulitkan pengembangan selanjutnya jika menggunakan database transaksional yang terpaku pada skema database yang ditetapkan di awal pengembangan aplikasi.

### 3. **Data citra/gambar menggunakan layanan Pihak Ketiga**

Sekarang telah banyak penyedia jasa *cloud computing* sebagai infrastruktur, seperti Amazon web Service, Google Cloud Storage. Alasan-alasan menggunakan AWS sebagai *data storage* untuk gambar adalah sebagai berikut :

- (a) Skalabilitas aplikasi lebih terjaga.

Dengan memisahkan penyimpanan antara gambar dan server sehingga lebih mudah *maintain*

perkembangan aplikasi, dan lebih fokus terhadap pengembangan aplikasi.

- (b) Menyediakan *built-in* keamanan, fleksibel dan efisiensi [3].

#### 4. Optimasi *assets*

Dalam banyak kesempatan, penulis seringkali mendapati bahwa *delay* untuk *loading assets* lebih lama daripada *loading data* dari database. Berikut penulis akan memaparkan hasil analisa berupa penyebab dan *tackling* permasalahan tersebut.

- (a) *Useless assets* yang disertakan dalam halaman: Memisahkan *essentials assets* dan menyertakan *script* yang hanya digunakan oleh halaman tersebut.
- (b) Logika penyusunan *script* yang tidak efektif dan optimal (misal: ada satu *script* yang menyertakan file yang tidak diperlukan): *pre-processing* berupa *minifying*, *optimization*, *compiling*, *compression* terhadap *assets*.
- (c) Latensi ke *server* yang cukup tinggi (misal: kecepatan sambungan internet yang rendah): *Caching*, *upgrading server* agar dapat "lebih terjangkau" secara jaringan, penerapan PWA (*Progressive web Apps*) untuk sisi *user experience*.

#### 3.1.5.7 Spesifikasi Kebutuhan Fungsional

Berdasarkan deskripsi umum sistem pada Subbab 3.1.1, maka dapat disimpulkan bahwa kebutuhan fungsional dari aplikasi lelang online, yang dipaparkan dalam Tabel 3.1.

**Tabel 3.1** Kebutuhan Fungsional Aplikasi Lelang *Online*

<b>No.</b>	<b>Parameter</b>	<b>Ketersediaan</b>
1	Manajemen Akun	Pengguna: Memperbarui dan mengubah informasi pengguna dalam akun
2	Manajemen Penawaran	Pengguna : Menawar barang, mendapat informasi barang, mencari barang yang diinginkan, dan fitur-fitur yang mempermudah pengguna dalam penawaran barang
3	Manajemen Barang Lelang	Pengguna:Mendaftarkan barang untuk dilelang, melihat progress kemajuan lelang
4	Manajemen Interaksi Antarpengguna	Pengguna: Berkirim pesan, memberi dan melihat <i>review</i> kepada pengguna lain, melaporkan pengguna atau barang
6	Manajemen Kupon	Pengelola: Membuat kupon, melihat daftar kupon, melihat riwayat penggunaan kupon

### 3.1.6 Spesifikasi Kebutuhan Non-Fungsional

Kebutuhan non-fungsional yang harus dipenuhi oleh aplikasi ini berhubungan dengan faktor-faktor sebagai berikut:

**Tabel 3.2** Kebutuhan Non-Fungsional Aplikasi Lelang *Online*

<b>No.</b>	<b>Parameter</b>	<b>Ketersediaan</b>
1	Ketersediaan	Aplikasi harus dapat berjalan pada <i>browser</i> , tanpa dibatasi waktu dan tempat selama browser tersebut tersambung ke jaringan internet.
2	Bahasa	Bahasa yang digunakan pada antarmuka merupakan bahasa Indonesia
3	Otorisasi	Setiap pengguna hanya berhak mengakses dan mengubah data yang merupakan milik pengguna tersebut.
4	Kecepatan	Rata-rata waktu akses halaman tidak boleh lebih dari 3 detik.
6	<i>Positive User Experience</i>	Aplikasi memberikan kesan positif terhadap pengalaman penggunaan aplikasi
7	<i>Security</i>	Koneksi terhadap <i>web</i> menggunakan SSL/ <i>https</i> .
.. dilanjutkan ke halaman selanjutnya		



Tabel 3.2 – lanjutan dari halaman sebelumnya

No.	Parameter	Ketersediaan
8	<i>Maintainability</i>	Aplikasi haruslah bersifat <i>maintainable</i> terhadap <i>developer</i> , dan tidak terlalu sensitif terhadap perubahan (jika terdapat perubahan fitur di masa depan, tidak harus <i>refactoring</i> atau mengubah struktur program secara keseluruhan).

### 3.1.7 Tugas dan Hak Akses Aktor

Aplikasi lelang *online* dapat digunakan sebagai wadah bagi para pecinta lelang *online* untuk melakukan kegiatan lelang atau bagi para penjual atau pembeli yang ingin menjual atau membeli barang yang sesuai baik dari kualitas maupun harga. Identifikasi aktor dalam sistem lelang *online* dijelaskan dalam Tabel 3.3, serta detail tugas dan hak akses pengguna dapat dilihat pada Tabel 3.4.

**Tabel 3.3** Identifikasi Aktor

No	Aktor	Keterangan
1	Pengguna	Pengguna yang dimaksud adalah pengguna yang menggunakan fungsionalitas lelang dalam sistem. Pengguna dapat menjadi seorang <i>bidder</i> ataupun seorang <i>auctioneer</i> , dimana pengguna dapat menjual barang untuk dilelang, dan dapat pula melelang barang
2	<i>Administrator</i>	Bertugas melihat dan mengawasi jalannya lelang, melihat laporan keluhan dari pengguna, serta melakukan <i>block</i> pengguna jika terdeteksi adanya tindakan yang tidak sesuai dengan kaidah lelang.

**Tabel 3.4** Detail Tugas dan Hak Akses

<b>Aktor</b>	<b>Tugas</b>	<b>Hak Akses</b>	<b>Kemampuan yang harus dimiliki</b>
Pengguna	Melakukan aktivitas lelang dan aktivitas interaksi antarpengguna	Manajemen barang & aktivitas lelang dan interaksi antarpengguna.	Paham penggunaan komputer, mengakses internet dengan <i>browser</i> dengan baik
<i>Administrator</i>	Bertugas mengawasi lelang, memonitor laporan keluhan dari pengguna, serta melakukan <i>block</i> pengguna jika diperlukan.	Mengakses halaman-halaman monitoring dari sistem terpisah.	Paham penggunaan komputer, mengakses internet lewat <i>browser</i> , memahami peraturan lelang dan <i>capable</i> untuk mengawasi lelang.

## 3.2 Perancangan Sistem

### 3.2.1 Spesifikasi Kasus Penggunaan

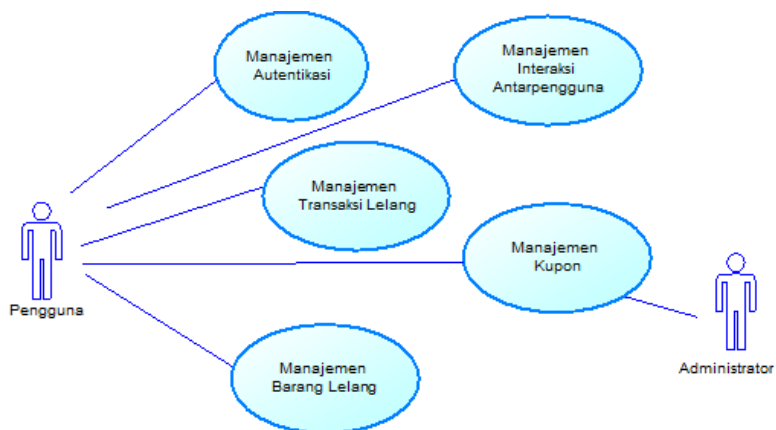
Kasus penggunaan dimaksudkan untuk menurunkan kebutuhan fungsional yang telah dispesifikasikan sebelumnya

pada Tabel 3.1.

Selanjutnya, akan dipaparkan masing-masing Spesifikasi Kasus Penggunaan untuk semua Kasus Penggunaan yang telah didefinisikan pada tabel 3.5.

**Tabel 3.5** Tabel Kasus Penggunaan

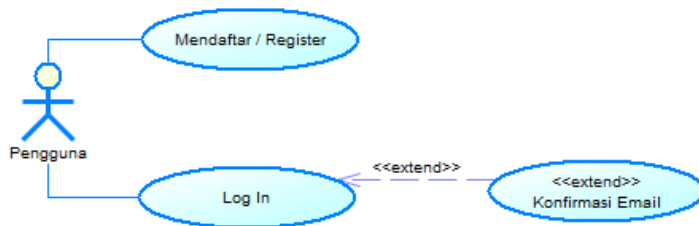
<b>ID Kasus Penggunaan</b>	<b>Kasus Penggunaan</b>
KP-01	Manajemen Authentikasi Pengguna
KP-02	Manajemen Transaksi Lelang
KP-03	Manajemen Barang Lelang
KP-04	Manajemen Interaksi Antar Pengguna
KP-05	Manajemen kupon



**Gambar 3.5** Diagram Kasus Penggunaan Aplikasi

### 3.2.1.1 KP01. Manajemen Autentikasi Pengguna

Pada kasus penggunaan ini, pengguna dapat memanajemen autentikasi dan pendaftaran ke dalam sistem.



**Gambar 3.6** Diagram Kasus Penggunaan Manajemen Autentikasi Pengguna

**Tabel 3.6** Spesifikasi Kasus Penggunaan Registrasi

<b>Kode</b>	UC-01.01
<b>Nama</b>	<b>Registrasi</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	-Pengguna mendaftar ke dalam akun agar masuk ke dalam sistem
<b>Tipe</b>	Fungsional
<b>Precondition</b>	Pengguna belum memiliki akun di aplikasi
<b>Postcondition</b>	Pengguna sudah memiliki akun terdaftar di aplikasi
<b>Alur Kejadian Normal</b>	
	<ol style="list-style-type: none"> <li>1. Pengguna membuka Halaman Registrasi</li> <li>2. Sistem menampilkan halaman yang berisi Form Registrasi</li> <li>3. Pengguna mengisi form tersebut</li> <li>4. Setelah selesai mengisi, pengguna mengklik tombol "Registrasi"</li> <li>5. Sistem memvalidasi data yang dimasukkan pengguna</li> <li>6. Jika data valid, sistem melakukan <i>redirect page</i> ke halaman <i>landing page</i> dalam keadaan sudah terautentikasi &amp; akun berhasil didaftarkan, dan sistem mengirimkan <i>email</i> konfirmasi <i>email</i> ke alamat <i>email</i> yang didaftarkan.</li> </ol>
<b>Alur Kejadian Alternatif</b>	
	<b>Data yang dimasukkan pengguna tidak valid</b>
	<ol style="list-style-type: none"> <li>5a. Sistem tidak dapat memvalidasi data yang dimasukkan pengguna.</li> <li>5b. Sistem melakukan <i>redirect</i> ke halaman form registrasi (langkah 2) dengan <i>error message</i>.</li> </ol>

Tabel 3.7 Spesifikasi Kasus Penggunaan *Login*

<b>Kode</b>	UC-01.02
<b>Nama</b>	<b><i>Login</i></b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Pengguna melakukan <i>login</i> agar dapat masuk ke dalam aplikasi dalam keadaan terautentikasi.
<b>Tipe</b>	Fungsional
<b><i>Precondition</i></b>	Pengguna masuk ke dalam aplikasi dalam keadaan belum terautentikasi
<b><i>Postcondition</i></b>	Pengguna masuk ke dalam aplikasi dalam keadaan sudah terautentikasi
<b>Alur Kejadian Normal</b>	
	<ol style="list-style-type: none"> <li>1. Pengguna membuka Halaman Login</li> <li>2. Sistem menampilkan halaman Login</li> <li>3. Pengguna mengisi halaman sesuai <i>credential</i> yang dimiliki</li> <li>4. Setelah selesai mengisi, pengguna mengklik tombol "Login"</li> <li>5. Sistem memverifikasi <i>credential</i> yang diberikan</li> <li>6. Jika data benar, sistem <i>redirect</i> ke halaman <i>landing page</i> dalam keadaan sudah terautentikasi.</li> </ol>
<b>Alur Kejadian Alternatif</b>	
	<b>Data yang dimasukkan pengguna tidak valid</b>
	<ol style="list-style-type: none"> <li>5a. Sistem tidak dapat memverifikasi <i>credential</i> pengguna.</li> <li>5b. Sistem <i>redirect</i> ke halaman Login (langkah 2) dengan <i>error message</i>.</li> </ol>

**Tabel 3.8** Spesifikasi Kasus Penggunaan: Konfirmasi *Email*

<b>Kode</b>	UC-01.03
<b>Nama</b>	<b>Konfirmasi Email</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Pengguna melakukan konfirmasi <i>email</i> agar status akun pengguna menjadi teraktivasi
<b>Tipe</b>	Fungsional
<b>Precondition</b>	Status akun pengguna masih belum terverifikasi
<b>Postcondition</b>	Status akun pengguna masih sudah terverifikasi
<b>Alur Kejadian Normal</b>	
	<ol style="list-style-type: none"> <li>1. Pengguna membuka halaman <i>inbox email</i> pengguna di <i>sistem email service</i> yang mereka gunakan.</li> <li>2. pengguna mencari dan membuka <i>email</i> konfirmasi yang dikirimkan oleh Lelangapa</li> <li>3. <i>Sistem email service</i> pengguna menampilkan isi <i>email</i> konfirmasi, beserta sebuah tombol "Konfirmasi email"</li> <li>4. Pengguna mengklik tombol "Konfirmasi Email"</li> <li>5. Halaman <i>browser</i> akan <i>redirect</i> ke URL konfirmasi email</li> <li>6. <i>Sistem</i> menampilkan halaman <i>landing page</i> dimana status akun pengguna sudah terverifikasi.</li> </ol>
<b>Alur Kejadian Alternatif</b>	
	-



### 3.2.1.2 KP02. Manajemen Transaksi Lelang

Pada kasus penggunaan ini, pengguna akan dapat memanajemen transaksi dan penawaran-penawaran yang ia berikan terhadap barang yang terdaftar dalam aplikasi.



**Gambar 3.7** Diagram Kasus Penggunaan Manajemen Traksaksi Lelang

**Tabel 3.9** Spesifikasi Kasus Penggunaan: Melihat Barang yang Dilelang

<b>Kode</b>	UC-02.01
<b>Nama</b>	<b>Melihat daftar barang yang dilelang</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Pengguna melihat daftar barang yang sedang dilelang
<b>Tipe</b>	Fungsional
<b>Pre Condition</b>	Sistem belum menampilkan daftar barang yang sedang dilelang
<b>Post Condition</b>	Sistem menampilkan daftar barang yang sedang dilelang
<b>Alur Kejadian Normal</b>	
	<ol style="list-style-type: none"> <li>1. Pengguna mengklik <i>icon</i> aplikasi di kiri atas halaman</li> <li>2. Sistem menampilkan halaman depan yang berisi daftar barang yang sedang dilelang  <i>Ket : Pada halaman depan, ditampilkan barang sesuai dengan kategori berdasarkan waktu dan popularitas, seperti Hot Item (barang yang paling ramai transaksi bidnya), Newest Item, dll.</i> </li> </ol>
<b>Alur Kejadian Alternatif</b>	
	-

**Tabel 3.10** Spesifikasi Kasus Penggunaan: Mencari Barang Lelang

<b>Kode</b>	UC-02.02
<b>Nama</b>	<b>Mencari Barang Lelang</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Pengguna ingin mencari barang lelang dengan kriteria nama tertentu
<b>Tipe</b>	Fungsional
<b>Pre Condition</b>	Pengguna menemukan barang lelang yang ia cari dengan kriteria <i>string</i> .
<b>Post Condition</b>	Pengguna menemukan barang lelang yang ia cari dengan kriteria <i>string</i> masukan.
<b>Alur Kejadian Normal</b>	
	<ol style="list-style-type: none"> <li>1. Pengguna memasukkan kriteria <i>string</i> pencarian di <i>field</i> masukan di <i>Header Bar</i></li> <li>2. Setelah selesai, pengguna mengklik tombol "Cari"</li> <li>3. Sistem mencari barang terdaftar yang sesuai dengan kriteria masukan pengguna</li> <li>4. Jika ketemu, sistem menampilkan halaman "Hasil Pencarian" beserta barang yang sesuai dengan kriteria pengguna.</li> <li>5. Pengguna lalu mengklik barang yang sesuai dengan keinginan</li> <li>6. Sistem menampilkan detail barang yang sesuai dengan keinginan pengguna</li> </ol>
<b>Alur Kejadian Alternatif</b>	
	<b>Tidak ada barang yang sesuai</b>
	<ol style="list-style-type: none"> <li>4 a. Sistem tidak dapat menemukan barang yang sesuai</li> <li>4 b. Sistem menampilkan "Hasil Pencarian" namun dengan keterangan "Hasil pencarian kosong"</li> </ol>

**Tabel 3.11** Spesifikasi Kasus Penggunaan: Mencari Barang Lelang

<b>Kode</b>	UC-02.03
<b>Nama</b>	<b>Mencari Barang Lelang</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Pengguna ingin mencari barang lelang dengan kriteria nama tertentu
<b>Tipe</b>	Fungsional
<b>Pre Condition</b>	Pengguna menemukan barang lelang yang ia cari dengan kriteria <i>string</i> .
<b>Post Condition</b>	Pengguna menemukan barang lelang yang ia cari dengan kriteria <i>string</i> masukan.
<b>Alur Kejadian Normal</b>	
	<ol style="list-style-type: none"> <li>1. Pengguna memasukkan kriteria <i>string</i> pencarian di <i>field</i> masukan di <i>Header Bar</i></li> <li>2. Setelah selesai, pengguna mengklik tombol "Cari"</li> <li>3. Sistem mencari barang terdaftar yang sesuai dengan kriteria masukan pengguna</li> <li>4. Jika ketemu, sistem menampilkan halaman "Hasil Pencarian" beserta barang yang sesuai dengan kriteria pengguna.</li> <li>5. Pengguna lalu mengklik barang yang sesuai dengan keinginan</li> <li>6. Sistem menampilkan detail barang yang sesuai dengan keinginan pengguna</li> </ol>
<b>Alur Kejadian Alternatif</b>	
	-

**Tabel 3.12** Spesifikasi Kasus Penggunaan: Melihat Riwayat Penawaran Lelang Barang

<b>Kode</b>	UC-02.04
<b>Nama</b>	<b>Melihat 3 Riwayat Penawaran Lelang Barang Teratas</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Pengguna ingin melihat riwayat penawaran lelang terhadap barang yang ia daftarkan
<b>Tipe</b>	Fungsional
<b><i>Pre Condition</i></b>	Pengguna menemukan barang lelang yang ia cari dengan kriteria <i>string</i> .
<b><i>Post Condition</i></b>	Pengguna menemukan barang lelang yang ia cari dengan kriteria <i>string</i> masukan.
<b>Alur Kejadian Normal</b>	
	<ol style="list-style-type: none"> <li>1. Pengguna membuka halaman "Kelola Barang"</li> <li>2. Pengguna mengklik barang yang ingin dilihat informasi riwayat penawaran lelangnya</li> <li>3. Sistem menampilkan halaman informasi barang tersebut</li> <li>4. Pengguna mengklik tombol "Lihat Penawaran Teratas"</li> <li>5. Sistem menampilkan <i>modal</i> berisi 3 riwayat penawaran lelang teratas barang tersebut.</li> </ol>
<b>Alur Kejadian Alternatif</b>	
	-

### 3.2.1.3 KP03. Manajemen Barang Lelang

Pada kasus penggunaan ini, pengguna akan dapat memanajemen barang yang ia daftarkan untuk dilelang, dan melihat proses monitoringnya, seperti yang dipaparkan pada penjelasan berikut.



**Gambar 3.8** Diagram Kasus Penggunaan Manajemen Barang Lelang

**Tabel 3.13** Spesifikasi Kasus Penggunaan: Mendaftarkan Barang Lelang

<b>Kode</b>	UC-03.01
<b>Nama</b>	<b>Mendaftarkan Barang Lelang</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Pengguna mendaftar barang untuk dilelang di dalam sistem
<b>Tipe</b>	Fungsional
<b><i>Precondition</i></b>	Barang yang akan dilelang belum terdaftar dalam sistem
<b><i>Postcondition</i></b>	Barang yang akan dilelang sudah terdaftar dalam sistem
<b>Alur Kejadian Normal</b>	
	<ol style="list-style-type: none"> <li>1. Pengguna dalam keadaan terautentikasi, mengklik "Item Anda" -&gt; "Add Items" pada <i>navbar</i> bagian atas halaman.</li> <li>2. Sistem menampilkan halaman yang berisi form pendaftaran barang</li> <li>3. Pengguna mengisi form tersebut sesuai data barang</li> <li>4. Setelah selesai mengisi, pengguna mengklik tombol "Daftar Barang"</li> <li>5. Sistem memvalidasi data yang dimasukkan pengguna</li> <li>6. Jika data valid, sistem <i>redirect</i> ke halaman "Kelola Barang" dalam keadaan barang baru sudah ditambahkan.</li> </ol>
<b>Alur Kejadian Alternatif</b>	
	<b>Data yang dimasukkan tidak valid</b>
	<ol style="list-style-type: none"> <li>5a. Sistem tidak dapat memvalidasi data yang dimasukkan pengguna.</li> <li>5b. Sistem <i>redirect</i> ke halaman form "Tambah Barang" (langkah 2) dengan <i>error message</i>.</li> </ol>

**Tabel 3.14** Spesifikasi Kasus Penggunaan: Mendaftarkan Barang Lelang

<b>Kode</b>	UC-03.02
<b>Nama</b>	<b>Memperbarui informasi barang yang dilelang</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Pengguna memperbarui informasi barang yang sebelumnya sudah terdaftar di dalam sistem
<b>Tipe</b>	Fungsional
<b>Precondition</b>	Informasi barang belum diperbarui.
<b>Postcondition</b>	Informasi barang sudah diperbarui.
<b>Alur Kejadian Normal</b>	
	<ol style="list-style-type: none"> <li>1. Pengguna dalam keadaan terautentikasi, mengklik "Item Anda" -&gt; "Manage Items" pada <i>navbar</i> bagian atas halaman.</li> <li>2. Sistem menampilkan halaman yang berisi daftar barang yang didaftarkan pengguna.</li> <li>3. Pengguna mengklik barang yang ingin diperbarui informasinya</li> <li>4. Sistem menampilkan halaman <i>form</i> "Perbarui barang".</li> <li>5. Pengguna mengisi informasi pembaruan barang di dalam form tersebut.</li> <li>6. Setelah selesai, pengguna mengklik tombol "Simpan Pembaruan".</li> <li>7. Sistem memvalidasi data (termasuk file gambar) yang dimasukkan pengguna, lalu sistem <i>redirect</i> ke halaman "Kelola Barang" dalam keadaan barang baru sudah ditambahkan.</li> </ol>
<b>Alur Kejadian Alternatif</b>	
	-



**Tabel 3.15** Spesifikasi Kasus Penggunaan: Melihat Barang yang Pernah Didaftarkan

<b>Kode</b>	UC-03.03
<b>Nama</b>	<b>Melihat Daftar Barang yang Pernah Dilelang</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Pengguna hendak melihat daftar semua barang yang pernah didaftarkan untuk dilelang di dalam sistem.
<b>Tipe</b>	Fungsional
<b><i>Precondition</i></b>	Informasi daftar barang belum ditampilkan.
<b><i>Postcondition</i></b>	Informasi daftar barang sudah ditampilkan.
<b>Alur Kejadian Normal</b>	
	<ol style="list-style-type: none"> <li>1. Pengguna dalam keadaan terautentikasi, mengklik "Item Anda" -&gt; "Manage Items" pada <i>navbar</i> bagian atas halaman.</li> <li>2. Sistem menampilkan halaman yang berisi daftar barang yang didaftarkan pengguna.</li> </ol>
<b>Alur Kejadian Alternatif</b>	
	-

**Tabel 3.16** Spesifikasi Kasus Penggunaan: Melihat Riwayat Penawaran Harga

<b>Kode</b>	UC-03.04
<b>Nama</b>	<b>Melihat Detail Riwayat Penawaran Harga</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Pengguna hendak melihat daftar semua barang yang pernah didaftarkan dalam sistem.
<b>Tipe</b>	Fungsional
<b>Precondition</b>	Informasi daftar barang belum ditampilkan.
<b>Postcondition</b>	Informasi daftar barang sudah ditampilkan.
<b>Alur Kejadian Normal</b>	
	<ol style="list-style-type: none"> <li>1. Pengguna dalam keadaan terautentikasi, mengklik "Item Anda" -&gt; "Manage Items" pada <i>navbar</i> bagian atas halaman.</li> <li>2. Sistem menampilkan halaman yang berisi daftar barang yang didaftarkan pengguna.</li> <li>3. Pengguna mengklik barang yang ingin dilihat daftar penawaran harganya</li> <li>4. Sistem menampilkan halaman detail informasi barang</li> <li>5. Pengguna mengklik tombol "Lihat Riwayat Penawaran"</li> <li>6. Sistem menampilkan halaman berisi daftar riwayat penawaran.</li> </ol>
<b>Alur Kejadian Alternatif</b>	
	-

### 3.2.1.4 KP04. Manajemen Interaksi Antarpengguna

Pada kasus penggunaan ini, pengguna difasilitasi untuk berinteraksi, memberikan *review*/testimoni terhadap pengguna lainnya sesuai dengan keinginan.



**Gambar 3.9** Diagram Kasus Penggunaan Manajemen Interaksi Antarpengguna

**Tabel 3.17** Spesifikasi Kasus Penggunaan: Melihat *Review* Pengguna

<b>Kode</b>	UC-04.01
<b>Nama</b>	<b>Melihat <i>Review</i> Pengguna</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Pengguna ingin melihat <i>review</i> pada pengguna tertentu
<b>Tipe</b>	Fungsional
<b><i>Precondition</i></b>	<i>Review</i> pengguna belum ditampilkan
<b><i>Postcondition</i></b>	<i>Review</i> pengguna berhasil ditampilkan
<b>Alur Kejadian Normal</b>	
	<ol style="list-style-type: none"> <li>1. Pengguna mengklik <i>link</i> profil pengguna</li> <li>2. Sistem menampilkan halaman profil pengguna</li> <li>3. Pengguna dapat melihat <i>review</i> pengguna di bagian kiri bawah beserta rata-rata <i>rating</i> yang diberikan.</li> </ol>
<b>Alur Kejadian Alternatif</b>	
	-

**Tabel 3.18** Spesifikasi Kasus Penggunaan: Memberikan *Review* Pengguna

<b>Kode</b>	UC-04.02
<b>Nama</b>	<b>Menambahkan <i>Review</i> Pengguna</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Pengguna ingin menambahkan <i>review</i> dari <i>transaksi</i> yang pernah dilakukan.
<b>Tipe</b>	Fungsional
<b><i>Precondition</i></b>	<i>Review</i> dari pengguna belum tercatat/tersimpan dalam sistem
<b><i>Postcondition</i></b>	<i>Review</i> dari pengguna berhasil tercatat dalam sistem
<b>Alur Kejadian Normal</b>	
	<ol style="list-style-type: none"> <li>1. Pengguna mengklik halaman 'Riwayat Transaksi'</li> <li>2. Sistem menampilkan halaman Riwayat Transaksi yang pernah dilakukan pengguna</li> <li>3. Pengguna mengklik <i>tab</i> jenis transaksi yang pernah dilakukan (Beli atau Lelang)</li> <li>4. Sistem menampilkan riwayat transaksi sesuai dengan jenis transaksi yang dipilih pengguna</li> <li>5. Pengguna mengklik transaksi yang ingin diberikan <i>review</i></li> <li>6. Sistem mengecek apakah <i>review</i> sudah pernah diberikan sebelumnya, lalu menampilkan form penilaian selanjutnya.</li> <li>7. Pengguna memasukkan jumlah <i>rating</i> dan deskripsi <i>rating</i>, lalu klik 'Simpan Review'</li> <li>8. Sistem memvalidasi masukan, jika valid sistem mengembalikan modal status sukses.</li> </ol>
<b>Alur Kejadian Alternatif</b>	
	-

**Tabel 3.19** Spesifikasi Kasus Penggunaan: Mengirimkan Pesan

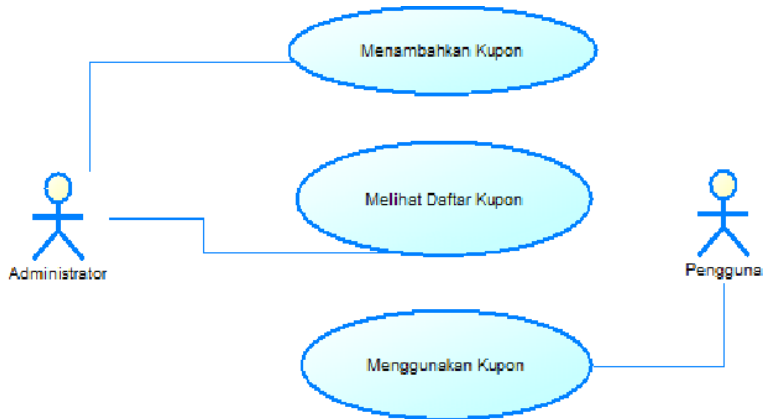
<b>Kode</b>	UC-04.04
<b>Nama</b>	<b>Mengirim Pesan</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Pengguna akan mengirimkan pesan kepada pengguna lainnya
<b>Tipe</b>	Fungsional
<b><i>Precondition</i></b>	Pesan yang dikirimkan pengguna belum tersimpan pada sistem
<b><i>Postcondition</i></b>	Pesan yang dikirimkan pengguna berhasil tersimpan pada sistem
<b>Alur Kejadian Normal</b>	
	<ol style="list-style-type: none"> <li>1. Pengguna mengklik <i>URL</i> pengguna tujuan yang ingin dikirim pesan</li> <li>2. Sistem menampilkan halaman profil pengguna tujuan</li> <li>3. Pengguna mengklik tombol "Kirim Pesan"</li> <li>4. Sistem menampilkan halaman percakapan dengan pengguna tersebut.</li> <li>5. Pengguna memasukkan pesan yang ingin dikirimkan. lalu klik tombol 'Kirim'</li> <li>6. Sistem mengirimkan pesan, lalu sistem kembali menampilkan halaman pengguna dengan informasi pesan yang sudah terkirim muncul di riwayat percakapan pengguna dengan pengguna tujuan</li> </ol>
<b>Alur Kejadian Alternatif</b>	
	-

**Tabel 3.20** Spesifikasi Kasus Penggunaan: Melihat & Membaca Pesan

<b>Kode</b>	UC-04.05
<b>Nama</b>	<b>Melihat daftar Pesan</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Pengguna ingin melihat daftar percakapan/ daftar perpesanan yang pernah dilakukan pengguna
<b>Tipe</b>	Fungsional
<b><i>Precondition</i></b>	Daftar percakapan/ daftar perpesanan belum ditampilkan
<b><i>Postcondition</i></b>	Daftar percakapan/ daftar perpesanan berhasil ditampilkan
<b>Alur Kejadian Normal</b>	
	<ol style="list-style-type: none"> <li>1. Pengguna mengklik tombol <i>Conversations</i> di <i>navbar</i> aplikasi</li> <li>2. Sistem menampilkan halaman daftar percakapan pengguna</li> <li>3. Pengguna mengklik percakapan yang ingin dilihat/dibaca</li> <li>4. Sistem menampilkan detail percakapan yang diklik sebelumnya.</li> </ol>
<b>Alur Kejadian Alternatif</b>	
	-

### 3.2.1.5 KP05. Manajemen Voucher

Kasus penggunaan ini seluruhnya digunakan oleh *administrator* aplikasi dan dilakukan di sistem terpisah. Kasus penggunaan ditujukan untuk mempermudah *administrator* dalam memanajemen kupon yang dibagikan oleh pengguna.



**Gambar 3.10** Diagram Kasus Penggunaan Manajemen Kupon



**Tabel 3.21** Spesifikasi Kasus Penggunaan: Menambahkan Kupon

<b>Kode</b>	UC-05.01
<b>Nama</b>	<b>Menambahkan Kupon</b>
<b>Aktor</b>	<i>Administrator</i>
<b>Deskripsi</b>	<i>Administrator</i> ingin menggunakan kupon/voucher yang ia miliki untuk pada sebuah transaksi
<b>Tipe</b>	Fungsional
<b><i>Precondition</i></b>	Kupon baru belum berhasil tersimpan di sistem
<b><i>Postcondition</i></b>	Kupon baru berhasil tersimpan di sistem
<b>Alur Kejadian Normal</b>	
	<ol style="list-style-type: none"> <li>1. <i>Administrator</i> membuka halaman 'Tambah Kupon'</li> <li>2. Sistem menampilkan halaman <i>form</i> Tambah Kupon</li> <li>3. <i>Administrator</i> memasukkan informasi kupon baru yang akan ditambahkan</li> <li>4. Sistem mengecek permintaan penggunaan kupon, jika permintaan dapat diverifikasi dan valid, sistem melakukan <i>redirect</i> ke halaman manajemen kupon.</li> </ol>
<b>Alur Kejadian Alternatif</b>	
	-

**Tabel 3.22** Spesifikasi Kasus Penggunaan: Mendaftarkan Barang Lelang

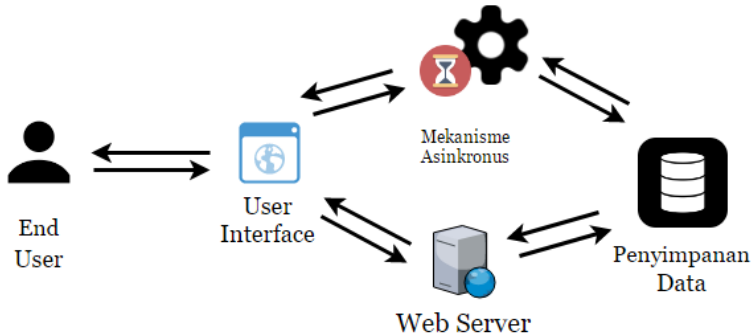
<b>Kode</b>	UC-05.02
<b>Nama</b>	<b>Memasukkan Kupon pada Transaksi</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Pengguna ingin menggunakan kupon/voucher yang ia miliki untuk pada sebuah transaksi
<b>Tipe</b>	Fungsional
<b>Precondition</b>	Pengguna belum berhasil melakukan <i>submit</i> kode kupon ke dalam transaksi barang
<b>Postcondition</b>	Pengguna berhasil melakukan <i>submit</i> kode kupon ke dalam transaksi barang
<b>Alur Kejadian Normal</b>	
	<ol style="list-style-type: none"> <li>1. Pengguna membuka halaman 'Riwayat Transaksi Lelang'</li> <li>2. Sistem menampilkan halaman Riwayat Transaksi Lelang pengguna</li> <li>3. Pengguna mengklik tombol 'Masukkan Kupon' pada transaksi yang diinginkan</li> <li>4. Sistem mengecek permintaan penggunaan kupon</li> <li>5. Jika permintaan dapat diverifikasi dan valid, sistem menampilkan <i>modal</i> berisi <i>input field</i> kupon</li> <li>6. Pengguna memasukkan kupon yang ingin dimasukkan, lalu mengklik tombol 'Submit'</li> <li>7. Sistem memvalidasi kupon dan status barang</li> <li>8. Jika valid, sistem menerapkan penggunaan kupon ke dalam transaksi barang</li> <li>9. Sistem lalu menampilkan <i>modal</i> yang berisi informasi sukses penggunaan kupon pada transaksi</li> </ol>
<b>Alur Kejadian Alternatif</b>	
	-

**Tabel 3.23** Spesifikasi Kasus Penggunaan: Menambahkan Kupon

<b>Kode</b>	UC-06.03
<b>Nama</b>	<b>Melihat Daftar Kupon</b>
<b>Aktor</b>	Administrator
<b>Deskripsi</b>	Administrator hendak melihat kupon yang terdaftar dalam sistem.
<b>Tipe</b>	Fungsional
<b><i>Precondition</i></b>	Daftar kupon belum ditampilkan
<b><i>Postcondition</i></b>	Daftar kupon berhasil ditampilkan
<b>Alur Kejadian Normal</b>	
	<ol style="list-style-type: none"> <li>1. <i>Administrator</i> membuka halaman 'Manajemen Kupon'</li> <li>2. Sistem menampilkan daftar kupon yang tersimpan dalam sistem</li> </ol>
<b>Alur Kejadian Alternatif</b>	
	-

**Tabel 3.24** Spesifikasi Kasus Penggunaan: Mendaftarkan Barang Lelang

<b>Kode</b>	UC-06.04
<b>Nama</b>	<b>Memasukkan Kupon pada Transaksi</b>
<b>Aktor</b>	Pengguna
<b>Deskripsi</b>	Pengguna ingin menggunakan kupon/voucher yang ia miliki untuk pada sebuah transaksi
<b>Tipe</b>	Fungsional
<b>Precondition</b>	Pengguna belum berhasil mensubmit kode kupon ke dalam transaksi barang
<b>Postcondition</b>	Pengguna berhasil mensubmit kode kupon ke dalam transaksi barang
<b>Alur Kejadian Normal</b>	
	<ol style="list-style-type: none"> <li>1. Pengguna membuka halaman 'Riwayat Transaksi Lelang'</li> <li>2. Sistem menampilkan halaman Riwayat Transaksi Lelang pengguna</li> <li>3. Pengguna mengklik tombol 'Masukkan Kupon' pada transaksi yang diinginkan</li> <li>4. Sistem mengecek permintaan penggunaan kupon</li> <li>5. Jika permintaan dapat diverifikasi dan valid, sistem menampilkan <i>modal</i> berisi <i>input field</i> kupon</li> <li>6. Pengguna memasukkan kupon yang ingin dimasukkan, lalu mengklik tombol 'Submit'</li> <li>7. Sistem memvalidasi kupon voucher dan status barang</li> <li>8. Jika valid, sistem menerapkan penggunaan kupon ke dalam transaksi barang</li> <li>9. Sistem lalu menampilkan <i>modal</i> yang berisi informasi sukses penggunaan kupon pada transaksi</li> </ol>
<b>Alur Kejadian Alternatif</b>	
	-



**Gambar 3.11** Arsitektur Dasar Aplikasi

### 3.2.2 Identifikasi Komponen Fundamental

Berdasarkan Bab Analisa, dapat diidentifikasi dan divisualisasikan pada Gambar 3.11 yaitu komponen-komponen penting dalam pembuatan aplikasi sebagai berikut:

1. *Webserver*
2. Mekanisme penyimpanan data (*database dan data storage*)
3. *User Interface* sebagai media terhadap *end-user*
4. Mekanisme Asinkronus untuk mengakomodasi fitur *realtime*

### 3.2.3 Technology Options

Pada subbab sebelumnya, penulis sudah memaparkan arsitektur dasar yang dibutuhkan dalam rancang bangun aplikasi. Terkait dengan arsitektur dasar tersebut, banyak pilihan teknologi yang dapat mengimplementasikan arsitektur tersebut. Dalam pemaparan selanjutnya, akan dijelaskan alasan penulis *best practice* dalam pemilihan teknologi yang digunakan, didasarkan pada *best practices* dan pengalaman-pengalaman penulis.

Keterkaitan dengan aspek-aspek yang dijelaskan sebelumnya pada Subbab 3.1.4.

### 3.2.3.1 NGINX sebagai *web* Server

- Kelebihan
  1. Konfigurasi yang lebih *friendly* dan terstruktur
  2. Ketersediaan fitur yang lengkap & krusial (*reverse proxy*, memungkinkan skalabilitas & *load balancing*)
  3. *Learning-gap* yang kecil terhadap pengalaman penulis/sudah familiar
- Opsi lainnya
  1. Apache2: Fiturnya kurang lengkap
  2. Node.js: *Learning-gap* yang besar bagi penulis/belum familiar
  3. Python: Belum Familiar, dan perlu eksplorasi fitur lebih dalam

### 3.2.3.2 POSTGRESQL untuk Penyimpanan Data

- Kelebihan
  1. *Learning gap* yang kecil
  2. Stabil karena telah digunakan dan dikembangkan oleh banyak *developer* selama bertahun-tahun
- Opsi lainnya
  1. SQL Server: Instalasi yang kompleks, penggunaan *resource* yang cukup besar

### 3.2.3.3 MONGODB untuk Penyimpanan Data Nontansaksional

- Kelebihan
  1. *Learning curve* yang mudah/sintaksnya kurang lebih sama dengan sintaks *database* transaksional pada umumnya

2. Performa yang cepat karena menggunakan BSON
  3. Fitur yang lengkap untuk *sustainability* aplikasi seperti (Replikasi, Sharding, dll)
  4. *Handling* terhadap data yang sangat besar yang cukup bagus, cocok untuk data yang masif seperti *chatting*.
- Opsi lainnya
    1. Redis: Cepat, namun penyimpanan dilakukan di RAM sehingga lebih cocok untuk penyimpanan *auth session*, bukan untuk penyimpanan data yang sifatnya masif
    2. Cassandra: *Learning-gap* yang besar, namun fiturnya lengkap untuk *data mining*

#### 3.2.3.4 CDN sebagai *Assets Sources*

- Kelebihan
  1. Akses cepat karena besar kemungkinan asset tersebut telah *dicache* sebelumnya dalam browser pengguna
  2. Mengurangi *bandwith* server
  3. Telah dioptimasi oleh pengembang masing-masing asset.
- Opsi lainnya
  1. Disimpan dalam server: Mengurangi *bandwith* server (*cost* meningkat)

#### 3.2.3.5 AWS S3 untuk *Content Growth Scalability*

- Kelebihan
  1. *Benefit* yang sangat *krusial*: keamanan, skalabilitas, *availability* - karena sudah dihandle langsung oleh pengembang *cloud computing* yang ahli di bidangnya
  2. Perkembangan jumlah konten yang akan disimpan (gambar barang yang diupload pengguna) tentunya bersifat sangat masif, sehingga tidak mungkin disimpan dalam server

- Opsi lainnya
  1. Disimpan dalam server: Mengurangi performa server karena sifatnya yang memakan *resource* cukup banyak, dan menambah *cost* untuk *upgrade server storage*

### 3.2.3.6 SENDGRID untuk SMTP Relay

- Kelebihan
  1. Konfigurasi yang mudah
  2. Dokumentasi yang cukup lengkap dan mudah ditemukan
  3. Fitur yang lengkap
  4. Adanya *free storage* dari akun Github Student Pack penulis
- Opsi lainnya
  1. MailChimp: Dokumentasi kurang lengkap, tidak ada *free storage* untuk akun penulis

### 3.2.3.7 VUE.JS untuk *Workloads Sharing*

- Kelebihan
  1. *Learning-gap* relatif kecil dibandingkan *Javascript tools* lainnya, karena didesain khusus untuk Laravel
  2. Adanya program utilitas (webpack) yang membuat performa Vue.js jauh lebih cepat
  3. Logika aplikasi dapat diobfuscate dengan webpack (*embedded* dalam Laravel)
- Opsi lainnya
  1. React: *Learning gap* dan *learning curve* yang sangat besar untuk penulis
  2. jQuery: tidak efektif karena *code smells* yang ditimbulkan cukup banyak



### 3.2.3.8 SOCKET.IO untuk Mekanisme Asinkronus

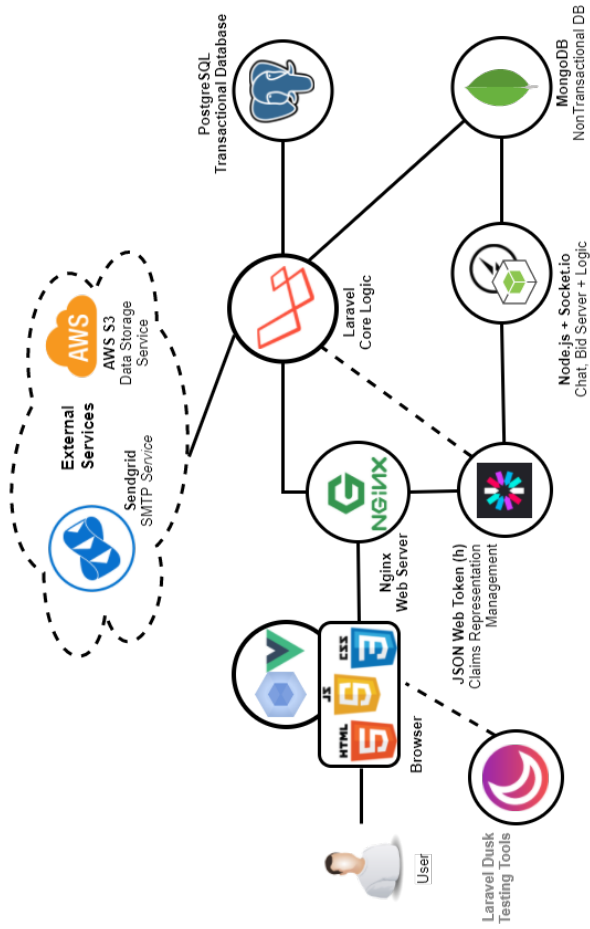
- Kelebihan
  1. *Learning-gap* relatif kecil dibandingkan *Javascript tools* lainnya, karena didesain khusus untuk Laravel
  2. Adanya program utilitas (webpack) yang membuat performa Vue.js jauh lebih cepat
  3. Logika aplikasi dapat *diobfuscate* dengan webpack (*embedded* dalam Laravel)
- Opsi lainnya
  1. React: *Learning gap* dan *learning curve* yang sangat besar untuk penulis
  2. jQuery: tidak efektif karena *code smells* yang ditimbulkan cukup banyak

### 3.2.3.9 JWT untuk Keamanan Soket

- Kelebihan
  1. *Library support* yang lengkap untuk komponen-komponen lainnya
  2. Efektif dan efisien karena tidak ada *query* ke database untuk autentikasi
- Opsi lainnya
  1. *Query* ke *database* secara konvensional: Sifat koneksi soket yang masif akan sangat memberatkan *database* jika setiap kali ada koneksi baru, harus melakukan *query database* sehingga tidak efektif
  2. *Session caching* dengan Redis: *Learning gap* yang besar

### 3.2.4 Arsitektur Perangkat Lunak

Arsitektur perangkat lunak dalam pengembangan aplikasi lelang *online* ini digambarkan pada diagram arsitektur pada Gambar 3.12.



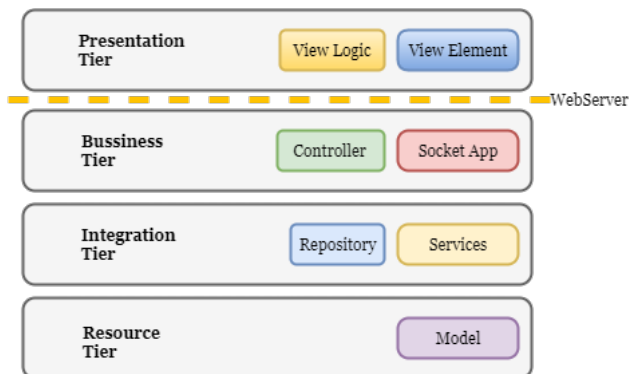
Gambar 3.12 Arsitektur dan Teknologi Aplikasi

### 3.2.5 Struktur Aplikasi

#### 3.2.5.1 *Presentation Tier*

*Presentation tier* adalah gabungan komponen yang bertanggung jawab terhadap *user interface* ke secara keseluruhan. Dalam *tier* ini menjadi 2 komponen, yaitu:

1. **Komponen *View Element*** bertanggungjawab terhadap dari elemen-elemen tampilan (HTML, CSS, JS), juga terkait dengan *templating* (pada penerapannya menggunakan *Blade templating*, yang sudah secara default masuk ke dalam *package* Laravel).
2. **Komponen *View Logic*** bertanggungjawab terhadap logika *view*, misal pada halaman melelang barang, maka *view logic* adalah logika program saat pengguna memasukkan penawaran harga, mengklik tombol "Tawar" hingga akhirnya penawaran tersebut sditerima/diproses oleh aplikasi.



**Gambar 3.13** Komponen Penyusun Struktur Aplikasi

### 3.2.5.2 *Bussiness Tier*

*Bussiness tier* adalah gabungan komponen yang bertanggung jawab kelangsungan proses bisnis. Dalam *tier* ini, terdapat 2 komponen yaitu:

1. **Komponen *Controller*** bertanggungjawab logika/proses bisnis aplikasi secara keseluruhan. *Controller* mengatur autentikasi, otorisasi, penerusan dan pemrosesan *request* dari pengguna kepada komponen pemroses seperti *data management layer* atau *external service layer*, lalu mengembalikan hasilnya kepada pengguna.
2. **Komponen *Socket App*** ditulis dengan menggunakan Node.js, dan komponen ini bertanggungjawab penuh untuk menyediakan *realtime connection* pada fitur berkiriman pesan dan proses *bidding*.

### 3.2.5.3 *Integration Tier*

*Integration tier* adalah gabungan komponen yang bertanggung jawab integrasi pemrosesan data dan *external service usages*. Dalam *tier* ini, terdapat komponen-komponen berikut:

1. **Komponen *Repository*** bertanggungjawab terhadap *data management*, dimana *controller* hanya bertugas meneruskan dan mengotorisasi *request*, maka *repository* bertanggungjawab untuk mengolah data tersebut sesuai permintaan. Lalu, *repository* bertanggungjawab meneruskan status proses (jika sifatnya keberhasilan, misal: *create data pengguna*), atau meneruskan hasil proses (misal: *request halaman tampilan barang*, maka *repository* bertanggungjawab meneruskan hasil *query* kepada *controller*).
2. **Komponen *Services*** bertanggungjawab penuh terhadap

akses aplikasi ke *external service* seperti *SMTP Relay* yang menggunakan *SendGrid*.

#### 3.2.5.4 *Resource/Data Access Tier*

*Resource tier* adalah gabungan komponen yang bertanggungjawab terhadap akses ke *database* dan *storage devices*, yaitu pengaksesan dan *preprocessing* data menjadi *suitable format* bagi komponen lainnya.

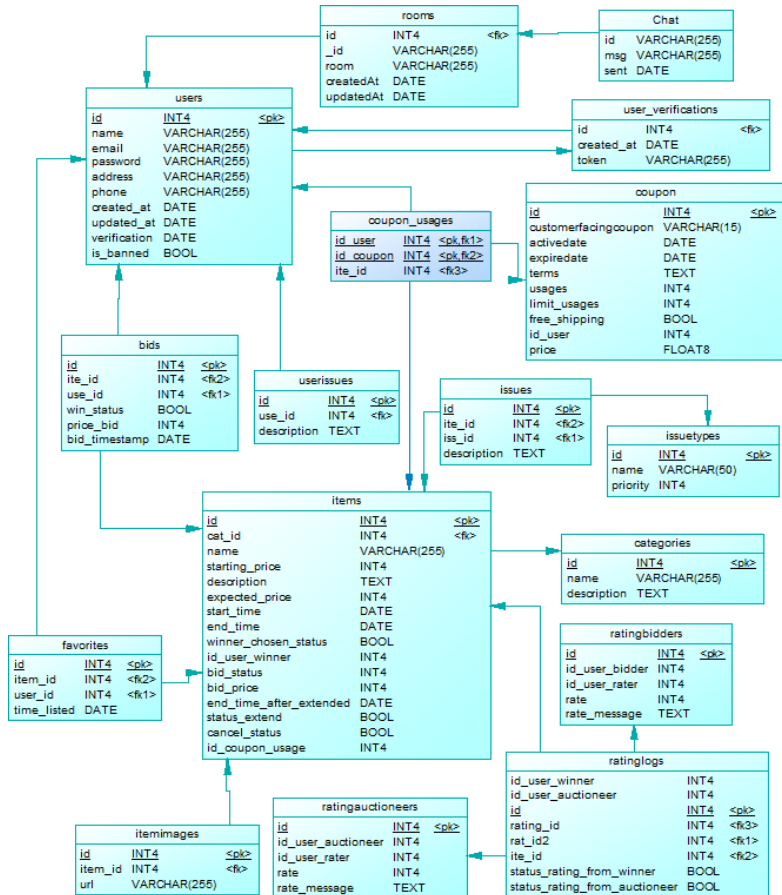
1. **Komponen *Model*** bertanggungjawab terhadap sebuah tabel/entitas dalam *database* aplikasi. Tidak hanya itu, *model* juga bertanggungjawab terhadap *preprocessing* informasi, yaitu pengubahan data dari *database* menjadi *collection/Eloquent ORM* yang merupakan *suitable format* bagi pemrosesan data di tier *repository* maupun *bussiness*.

#### 3.2.6 *Perancangan Database*

Pada subbab ini akan dijelaskan bagaimana rancangan basis data yang digunakan pada aplikasi lelang *online* ini. Sistem basis data yang digunakan pada aplikasi ini menggunakan dua jenis database, yaitu transaksional (menggunakan PostgreSQL) dan nontransaksional (menggunakan MongoDB). *Conceptual Data Model* (CDM) dan *Physical Data Model* (PDM) dari basis data sistem ini dijelaskan pada Gambar 3.14 dan Gambar 3.15.



### 3.2.6.2 Physical Database Model



Gambar 3.15 Physical Database Model (PDM) Aplikasi

### 3.2.7 Kamus Data *Database* Transaksional

Subbab ini akan berisi pemaparan detail tentang kamus data dalam *database* yang bersifat transaksional dengan total keseluruhan 15 tabel.

#### 3.2.7.1 Kamus data tabel *administrator*

**Tabel 3.25** Kamus Data Tabel *administrators*

No	Nama Atribut	Tipe Data	Keterangan
1	id	int	PK , <i>Autoincrement</i>
2	name	varchar(255)	Nama
3	username	varchar(255)	Username yang digunakan
4	password	varchar(255)	<i>Password</i>
5	created_at	timestamp	<i>Timestamp</i> default laravel
6	updated_at	timestamp	<i>Timestamp</i> default laravel
7	email	varchar(255)	<i>Email</i>

#### 3.2.7.2 Kamus data tabel *bids*

**Tabel 3.26** Kamus Data Tabel *bids*

No	Nama Atribut	Tipe Data	Keterangan
1	id	bigint	PK , <i>Autoincrement</i>
Dilanjutkan ke halaman selanjutnya			



Tabel 3.26 – lanjutan dari halaman sebelumnya

No	Nama Atribut	Tipe Data	Keterangan
2	id_user	int	FK ID <i>user</i> yang memberikan penawaran
3	id_item	bigint	FK ID barang yang ditawarkan
4	price_bid	bigint	Harga yang ditawarkan
5	win_status	bool	Status apakah penawaran ini memenangkan lelang
6	bid_timestamp	timestamp	<i>Timestamp</i>

### 3.2.7.3 Kamus data tabel *categories*

Tabel 3.27 Kamus Data Tabel *categories*

No	Nama Atribut	Tipe Data	Keterangan
1	id	bigint	PK, <i>Autoincrement</i>
2	name	int	Nama Kategori
3	description	bigint	Deskripsi Kategori

### 3.2.7.4 Kamus data tabel *coupon*

**Tabel 3.28** Kamus Data Tabel *coupon*

No	Nama Atribut	Tipe Data	Keterangan
1	coupon_id	int	PK, <i>Autoincrement</i>
2	<u>customerfacing</u> coupon	varchar(15)	String kupon yang diberikan kepada pengguna
3	activedate	timestamp	Tanggal aktif kupon
4	expireddate	timestamp	Tanggal nonaktif kupon
5	terms	varchar(255)	Deskripsi aturan penggunaan kupon
6	usages	bigint	Banyak penggunaan kupon
7	limit_usages	bigint	Batas penggunaan kupon
8	free_shipping	bool	Status apakah kupon berupa kupon <i>free shipping</i> atau tidak
9	price	float	Persentase diskon yang diberikan

**3.2.7.5 Kamus data tabel *coupon\_usages***

**Tabel 3.29** Kamus Data Tabel *coupon\_usages*

No	Nama Atribut	Tipe Data	Keterangan
1	id	int	PK, <i>Autoincrement</i>
2	id_user	bigint	FK ID <i>user</i> yang menggunakan kupon
3	id_item	bigint	FK ID <i>item</i>
4	id_coupon	int	FK ID kupon yang digunakan
5	price_after	bigint	FK harga setelah penggunaan kupon

### 3.2.7.6 Kamus data tabel *favorites*

**Tabel 3.30** Kamus Data Tabel *favorites*

No	Nama Atribut	Tipe Data	Keterangan
1	id	bigint	PK, <i>Autoincrement</i>
2	id_user	int	FK ID <i>user</i>
3	id_item_favorite	bigint	FK ID <i>item</i>
4	remove_status	bool	
5	time_listed	timestamp	Timestamp

### 3.2.7.7 Kamus data tabel *issues*

Tabel 3.31 – lanjutan dari halaman sebelumnya

No	Nama Atribut	Tipe Data	Keterangan
----	--------------	-----------	------------

Tabel 3.31 Kamus Data Tabel *issues*

No	Nama Atribut	Tipe Data	Keterangan
1	id	int	PK, <i>Autoincrement</i>
2	id_issue_type	int	FK ID tipe <i>issue</i> yang dilaporkan
3	id_user	int	FK ID <i>user</i> yang melaporkan
4	id_issued_object	int	FK ID <i>item</i> yang dilaporkan
5	description	text	Deskripsi Laporan

### 3.2.7.8 Kamus data tabel *issue\_types*

Tabel 3.32 Kamus Data Tabel *issuetypes*

No	Nama Atribut	Tipe Data	Keterangan
1	id	int	PK, <i>Autoincrement</i>
2	name	varchar(50)	Nama tipe laporan
3	priority	int	Prioritas laporan

### 3.2.7.9 Kamus data tabel *items*

**Tabel 3.33** Kamus Data Tabel *items*

No	Nama Atribut	Tipe Data	Keterangan
1	id	bigint	PK, <i>Autoincrement</i>
2	id_user	int	FK ID <i>user</i>
3	name	varchar(255)	Nama barang
4	description	text	Deskripsi barang
5	starting_price	bigint	Harga awal lelang
6	expected_price	bigint	Harga akhir yang diinginkan
7	start_time	timestamp	Awal waktu lelang
8	end_time	timestamp	Akhir waktu lelang
9	id_category	int	FK - ID kategori barang
10	<u>winner_chosen_status</u>	bool	Status terpilih pemenang lelang
11	id_user_winner	int	FK - ID <i>user</i> pemenang lelang
12	bid_price	bigint	Harga penawaran lelang tertinggi pada barang tersebut
13	<u>end_time_after_extended</u>	timestamp	Waktu akhir <i>extended</i>
14	status_extend	int	Status extend waktu lelang barang

Tabel 3.33 – lanjutan dari halaman sebelumnya

No	Nama Atribut	Tipe Data	Keterangan
15	cancel_status	bool	Status pembatalan lelang
16	id_coupon_usage	int	FK - ID kupon yang digunakan pada barang tersebut

### 3.2.7.10 Kamus data tabel *rating\_auctioneers*

Tabel 3.34 Kamus Data Tabel *ratingauctioneers*

No	Nama Atribut	Tipe Data	Keterangan
1	id	int	PK, <i>Autoincrement</i>
2	<u>id_user</u> _auctioneer	varchar(50)	FK ID <i>user</i> penjual
3	id_user_rater	int	FK ID <i>user</i> pemberi review
6	rate	bigint	Jumlah <i>rating</i> yang diberikan
7	id_item	int	FK ID barang yang diberi <i>rating</i>
8	rate_message	text	Deskripsi <i>rating</i>

### 3.2.7.11 Kamus data tabel *ratingbidders*

**Tabel 3.35** Kamus Data Tabel *ratingbidders*

No	Nama Atribut	Tipe Data	Keterangan
1	id	int	PK, <i>Autoincrement</i>
2	id_user_bidder	varchar(50)	FK ID <i>user</i> Pelelang
3	id_user_rater	int	FK ID <i>user</i> pemberi review
6	rate	bigint	Jumlah <i>rating</i> yang diberikan
7	id_item	int	FK ID barang yang diberi <i>rating</i>
8	rate_message	text	Deskripsi <i>rating</i>

### 3.2.7.12 Kamus data tabel *ratinglogs*

**Tabel 3.36** Kamus Data Tabel *ratinglogs*

No	Nama Atribut	Tipe Data	Keterangan
1	id	int	PK, <i>Autoincrement</i>
2	id_user_winner	varchar(50)	FK ID <i>user</i> pemenang lelang
3	id_user_auctioneer	int	FK ID <i>user</i> penjual
4	id_item	int	FK ID barang yang diberi <i>rating</i>

Tabel 3.36 – lanjutan dari halaman sebelumnya

No	Nama Atribut	Tipe Data	Keterangan
5	<u>status_rating</u> _from_winner	int	Flag apakah transaksi sudah diberi <i>rating</i> oleh pemenang lelang
6	<u>status_rating</u> _from_auctioneer	bigint	Flag apakah transaksi sudah diberi <i>rating</i> oleh penjual barang
7	id_rating_bidder	int	FK ID <i>rating</i> di tabel <i>ratingauctioneers</i>
8	<u>id_rating</u> _auctioneer	text	FK ID <i>rating</i> di tabel <i>ratingbidders</i>

### 3.2.7.13 Kamus data tabel *userverifications*

Tabel 3.37 Kamus Data Tabel *userverifications*

No	Nama Atribut	Tipe Data	Keterangan
1	id_user	int	FK ID <i>user</i>
2	created_at	timestamp	<i>Timestamp</i>
3	token	varchar(255)	<i>Token</i> random yang dibuat oleh sistem

### 3.2.7.14 Kamus data tabel *userissues*



**Tabel 3.38** Kamus Data Tabel *userissues*

<b>No</b>	<b>Nama Atribut</b>	<b>Tipe Data</b>	<b>Keterangan</b>
1	id	int	PK, <i>Autoincrement</i>
2	id_user	int	FK - ID <i>user</i> yang melaporkan
3	id_issued_user	int	FK - ID User yang dilaporkan
4	description	text	Deskripsi laporan yang diberikan

### 3.2.7.15 Kamus data tabel *users*

**Tabel 3.39** Kamus Data Tabel *users*

<b>No</b>	<b>Nama Atribut</b>	<b>Tipe Data</b>	<b>Keterangan</b>
1	id	int	PK, <i>Autoincrement</i>
2	name	varchar(255)	Nama
3	email	varchar(50)	<i>Email</i>
4	password	varchar(255)	<i>Password</i>
5	address	text	Alamat
6	phone	varchar(50)	No. HP
7	created_at	timestamp	<i>Timestamp</i> <i>record</i> dibuat
8	updated_at	timestamp	<i>Timestamp</i> <i>record</i> terakhir diperbarui
9	username	varchar(20)	Username

Tabel 3.39 – lanjutan dari halaman sebelumnya

No	Nama Atribut	Tipe Data	Keterangan
10	verification	timestamp	Tanggal user terverifikasi - jika belum diverifikasi, isinya NULL
11	is_banned	timestamp	Status <i>banned</i> pengguna

### 3.2.8 Kamus Data *Database* Non-Transaksional

Subbab ini akan berisi pemaparan detail kamus data dalam *database* yang bersifat non-transaksional. Total keseluruhan terdapat 3 *collections*.

#### 3.2.8.1 Kamus Data *Collection* userchats

Tabel 3.40 Kamus Data *Collection* userchat

No	Nama Atribut	Tipe Data	Keterangan
1	id	-	<i>Hash generated key</i>
2	room	varchar	<i>Room chat</i>
3	sender	int	FK ID <i>user</i> pengirim sender
4	msg	int	FK ID <i>user</i> penerima pesan
5	sent	ISODate	<i>Timestamp</i> pesan diterima

#### 3.2.8.2 Kamus Data *Collection* chatroom

**Tabel 3.41** Kamus Data *Collection chatroom*

<b>No</b>	<b>Nama Atribut</b>	<b>Tipe Data</b>	<b>Keterangan</b>
1	id	-	<i>Hash generated key</i>
2	room	varchar	<i>Room chat</i>
3	id_user_1	int	FK ID user pengirim sender
5	created_at	ISODate	<i>Timestamp pesan pertama</i>
6	updated_at	ISODate	<i>Timestamp pesan terakhir</i>

### 3.2.8.3 Kamus Data *Collection itemimages*

**Tabel 3.42** Kamus Data *Collection itemimage*

<b>No</b>	<b>Nama Atribut</b>	<b>Tipe Data</b>	<b>Keterangan</b>
1	id	-	<i>Hash generated key</i>
2	item_id	int	FK - ID barang
3	url	int	URL gambar

*(Halaman ini sengaja dikosongkan)*

## BAB IV

### IMPLEMENTASI

Pada bab ini dibahas mengenai implementasi aplikasi, dibagi menjadi empat subbab yaitu :

1. Implementasi Perangkat Keras / *Deployment*
2. Implementasi Perangkat Lunak
3. Implementasi Antarmuka / *User Interface*

#### 4.1 Implementasi Perangkat Keras/ *Deployment*

*Deployment* aplikasi dilakukan secara *online*, dalam sebuah *virtual private server* yang dihost oleh *Digital Ocean*. Spesifikasi VPS yang digunakan adalah sebagai berikut:

1. **Hardware**
  - (a) CPU: Intel(R) Xeon(R) CPU E5-2630L v2 @ 2.40GHz
  - (b) *Operating System*:
  - (c) RAM: 512MB
  - (d) *Storage Space*: 20GB
2. **Operating System**
  - (a) *Architecture*: 64bit
  - (b) *Kernel Version*: Linux 4.4.0-75-generic x86 64
  - (c) *OS Version*: Ubuntu 16.04.2 LTS Xenial
3. **Networking Stats**
  - (a) Tersambung ke Internet: Ya
  - (b) IP Publik: Ya
  - (c) Alamat IP Publik (IPv4): 188.166.179.2
  - (d) *Average Download Speed*: 1371 Mbit/s
  - (e) *Average Upload Speed*: 860.12 Mbit/s
  - (f) DNS: Google
4. **Domain Stats**
  - (a) *HTTPS Support*: Yes
  - (b) *SSL certificate issued by*: Avast
  - (c) Domain: <https://Lelangapa.com>

- (d) *Testing-purpose subdomain*:  
https://testing.lelangapa.com
- (e) *Domain issued by*: Namecheap

## 4.2 Implementasi Perangkat Lunak

Pada saat instalasi dan konfigurasi beberapa komponen perangkat lunak, tidak tidak keseluruhan prosesnya berjalan dengan baik dalam sekali percobaan. Berikut paparan beberapa kesulitan dan penyelesaiannya.

### 4.2.1 Strategi *Deployment* NGINX

#### 1. CORS (*Cross Origin Resource Sharing*)

CORS adalah sebuah mekanisme yang memungkinkan sebuah *website* menggunakan *resources*(seperti skrip Javascript, *fonts*, dll) untuk diakses dari sumber lain selain *domain* aslinya. Secara teknis, CORS mendefinisikan *protokol/cara* browser dan server untuk berinteraksi otorisasi permintaan *resource* dari domain lain, dan juga lebih aman karena *developer* dapat mengontrol otorisasi tersebut (daripada mengizinkan semua permintaan).

#### *The Problem*

Masalah koneksi ke server lelang (yang berjalan pada domain yang sama, namun port yang berbeda) tidak dapat tersambung karena *error* berikut :



**Gambar 4.1** Error CORS pada *Console Browser*

### ***Insight***

Selama 2 minggu kurang lebih penulis mencoba cara dari situs Stackoverflow untuk konfigurasi *server* lelang agar otorisasi CORS dapat dilakukan, namun tidak ada hasil. Masalah ini terselesaikan setelah menggunakan fitur *reverse proxy* dari Nginx.

### ***Solution***

Strateginya adalah sebagai berikut :

- (a) *Server* lelang berjalan pada localhost port 3000.
- (b) Diatur sebuah *subdomain* khusus – misalkan A.domain.com
- (c) NGINX dikonfigurasi dimana semua *requests* menuju A.domain.com ke aplikasi *localhost* yang kita maksud di poin 1a
- (d) Selain meneruskan *requests*, NGINX juga akan meneruskan *reply* dari *server* lelang tersebut kepada *client/origin* yang meminta *request* tersebut.

## **4.2.2 Strategi Deployment VUE.JS**

### **1. *Package Dependencies***

Pada versi terbaru Laravel (5.4\*), Laravel secara *default* menyertakan *package* Laravel Mix - yaitu fitur untuk *compiling assets* dengan Webpack, dengan hasil akhir *compiled assets* (terutama *script* Javascript) yang eksekusinya jauh lebih cepat, karena menggunakan V8 – sebuah *engine* Javascript yang telah dioptimasi yang bersifat *just-in-time* (JIT) yang memproduksi *machine code* dari sebuah *script* Javascript lalu dieksekusi.

### ***Main Problem***

Masalah muncul saat versi Laravel yang digunakan untuk membangun aplikasi adalah versi (5.3) – dan jika

Laravelnya diupgrade, tidak ada jaminan bahwa *deprecated dependencies* (keadaan dimana sebuah *package* tidak disupport oleh versi terbaru) – yang berarti harus *refactoring code* yang pasti memakan waktu lama.

### ***Insights***

Penulis menganalisa perbedaan mendasar *package.json* antara Laravel 5.3 dan 5.4 adalah sebagai berikut:

- a. Basis : Perubahan basis yang awalnya Gulp menjadi Webpack
- b. *Dependencies* : Webpack ternyata menggunakan beberapa plugin tambahan yang tidak diakomodasi dalam *package.json* di versi 5.3
- c. *Run Script* : Terdapat beberapa perubahan signifikan terhadap *run script alias* di versi 5.4 - dibandingkan pada versi 5.3.
- d. *Additional Files* : Terdapat beberapa file konfigurasi tambahan agar proses kompilasi aset dapat berjalan dengan baik.

### ***Solution***

Penulis lalu mengoreksi dan *update package.json* dengan pendekatan *trial and error*, dan bisa terselesaikan dengan mengkonfigurasi *package.json* sesuai pada Kode Sumber IV.1.

## **2. *Dependencies Optimization Problem***

Setelah menulis beberapa *script* Vue, penulis menganalisa bahwa setiap *script* Vue ternyata mempunyai *dependencies* yang sama, yaitu axios, Promise, toastr dan vue. Setiap file Vue membutuhkan *scripts* seperti pada Kode Sumber IV.2. Hal ini mengakibatkan semua *file* vue yang *dicompile*



ukurannya cukup besar (kurang lebih sebesar 400KB), padahal sebenarnya di dalam setiap *file* tersebut sebenarnya ada yang sama. Hal ini tentu tidak efektif, karna sebenarnya hal-hal yang sama tersebut bisa dipisahkan, dan dijadikan *cache* sehingga *loading* halaman bisa jauh lebih cepat.

### ***Insight & Solution***

Setelah penulis berdiskusi di forum Slack, beberapa pengguna Vue menyarankan untuk *compile* keseluruhan *dependencies* yang digunakan kedalam satu *file* terpisah, dan hanya menulis logika Vue untuk setiap file Vue, Isi *file* webpack.mix.js (*file* yang *dcompile* oleh Webpack) seperti pada Kode Sumber IV.3.

```

1 { "private": true,
2   "scripts": {
3     "_comment" : "Lists of running npm
                     ↪ commands defined here"
4   },
5   "devDependencies": {
6     "axios": "^0.15.3",
7     "bootstrap-sass": "^3.3.7",
8     "cross-env": "^3.2.3",
9     "jquery": "^3.1.1",
10    "laravel-mix": "0.*",
11    "lodash": "^4.17.4",
12    "vue": "^2.1.10"
13  },
14  "dependencies": {
15    "vue-resource": "^1.3.1"
16  } }

```

**Kode Sumber IV.1** Script package.json Aplikasi

```

1  window.axios = require('axios');
2  window.toastr = require('toastr');
3  window. = require('toastr');
4  require('vue-resource');

```

**Kode Sumber IV.2** *Script Webpack yang Belum Dioptimasi*

```

1  /* dependencies all compiled into one single file
   ↪ */
2  mix.js('scripts/dependencies.js', 'public/js');
3
4  /* dependencies all compiled into one single file
   ↪ */
5  mix.js('scripts/favorites.js', 'public/js');
6  mix.js('scripts/other_vue_script.js', 'public/js'
   ↪ );

```

**Kode Sumber IV.3** *Script Webpack yang Dioptimasi*

```

1  <script
2    src="dependencies.js" >
3  </script>
4  <script
5    src="custom_page_script.js" >
6  </script>

```

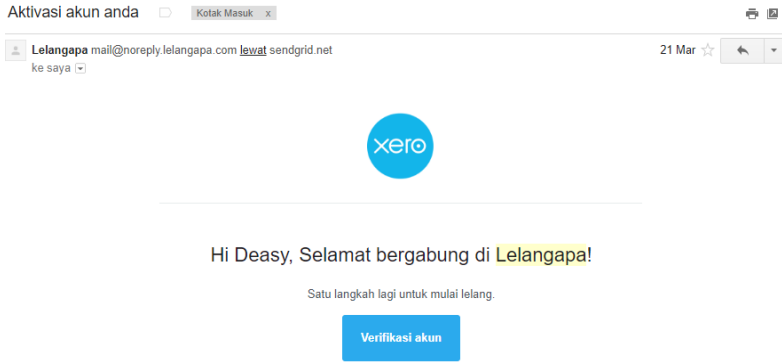
**Kode Sumber IV.4** *Including Script pada View*

#### 4.2.3 Strategi *Whitelisting* pada SENDGRID

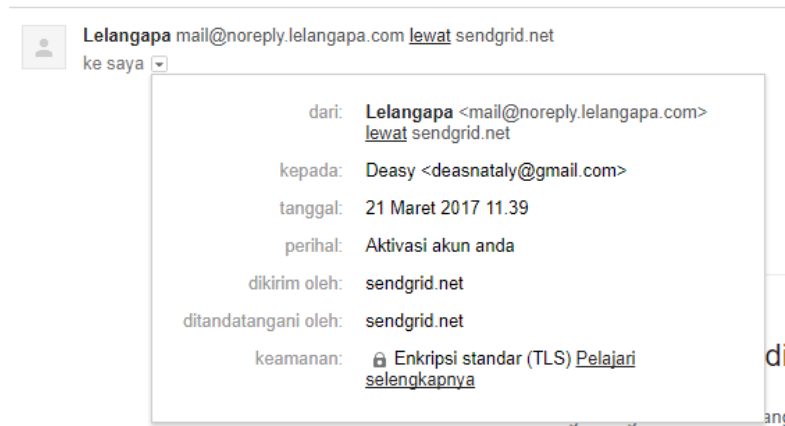
*Whitelisting* adalah kebalikan dari teknologi *blacklist*. Jika *blacklist* merupakan daftar dari sekumpulan *web* domain ataupun alamat email, dan URL yang terindikasi tidak “aman” sehingga secara otomatis akan diblokir oleh komputer maupun jaringan agar tidak dapat diakses, maka *whitelist* kebalikan dari *blacklist* yaitu daftar yang diperbolehkan untuk diakses oleh komputer atau jaringan dimana terdapat sekumpulan URL, *web* domain maupun alamat *email* yang “aman”.

Masalah dimulai saat *email* konfirmasi akun yang dikirimkan oleh sistem aplikasi lelang *online* - dengan menggunakan SMTP *relay* - kepada *email pengguna*, masuk ke dalam kotak pesan sebagai *spam*. Hal ini tentu tidak baik - karena seharusnya masuk ke *inbox* sebagaimana *email* pada umumnya. Setelah penulis mencoba mencari jalan keluar, terutama SendGrid tidak menyediakan *whitelisting* dengan menggunakan *root domain* (karena penulis membeli domain lelangapa.com, penulis tidak dapat mengirim *email* konfirmasi akun dengan menggunakan any\_email\_address@lelangapa.com, dan hingga buku ini ditulis penulis tetap tidak tahu alasan pastinya apa). Masalah ini baru selesai dengan menggunakan pemecahan berikut:

1. Membuat sebuah domain noreply.lelangapa.com
2. Meregister domain *whitelisting* di pengaturan SendGrid



**Gambar 4.2** *Whitelisting* Berhasil Dijalankan



**Gambar 4.3** Detail Informasi *Email* pada Kotak Masuk Pengguna

## 4.3 Implementasi Antarmuka / *User Interface*

### 4.3.1 Antarmuka Halaman Registrasi

Halaman ini dapat diakses oleh semua pengguna, baik yang belum terdaftar maupun sudah. Halaman ini menampilkan form berisi elemen *input* data diri, dan pengguna dapat mengisi lalu mengklik tombol daftar, dan untuk kasus alternatif dapat dilihat pada Tabel 3.6. Tidak ada *view logic* ataupun logika UI khusus dalam halaman ini. Kode sumber implementasi *back-end* dapat dilihat pada Kode Sumber IV.5.

Lelangapa Bid your items. Get your price.

---

Daftar

Nama

Nama anda

Email

Email anda

Password

Password

Konfirmasi Password

Konfirmasi Password

Alamat

Alamat anda

Phone

Phone number

Daftar

**Gambar 4.4** Halaman Antarmuka Registrasi

```

1  /*
2  * Menampilkan halaman register
3  * method : GET
4  */
5  public function showRegistrationForm(){
6      return view('auth.register2');
7  }
8
9  /*
10 * validator fungsi
11 */
12 protected function validator(array $data){
13
14     return Validator::make($data, [
15         'name' => 'required|max:255',
16         'email' => 'required|email|max
           ↳ :255|unique:users',
17         'password' => 'required|min:6|
           ↳ confirmed',
18         'username' => 'required|unique:
           ↳ users|min:5',
19         'phone' => 'numeric',
20     ]);
21 }
22
23 /*
24 * Dipanggil saat mengklik tombol daftar
25 * method : POST
26 */
27 public function register(Request $request){
28
29
30     /*      validasi data      */
31     $this->validator($request->all())->
           ↳ validate();
32
33     event(new Registered($user = $this->

```

```

34         ↪ create($request->all())));
35     $this->guard()->login($user);
36
37     /* notify activationService
38     to send activation mail to user's \textit
39     ↪ {email} */
40     $this->activationService->
41     ↪ sendActivationMail($user);
42
43     return $this->registered($request, $user)
44     ↪ ?
45     : redirect($this->redirectPath());
46 }

```

**Kode Sumber IV.5** Implementasi Antarmuka Registrasi

#### 4.3.2 Implementasi Halaman *Login*

Halaman ini dapat diakses oleh semua pengguna, baik yang belum terdaftar maupun sudah, dengan pengecualian pengguna tidak dalam keadaan sudah *login*. Halaman ini menampilkan *form* berisi elemen *input email* dan *password*, dan tombol *login*. Kasus normal dan alternatif dapat dilihat pada Tabel 3.7. Tidak ada *view logic* dalam halaman ini. Kode sumber implementasi *back-end* dapat dilihat pada Kode Sumber IV.6.

## Lelangapa Bid your items. Get your price.

**Gambar 4.5** Halaman Antarmuka *Login*

```

1 public function showLoginForm() {
2     /*Menampilkan halaman login; Method : GET
3         ↪ */
4     return view('auth.login2');
5 }
6 public function login(Request $request) {
7     /* Setelah klik tombol login
8         Method : POST */
9     $this->validateLogin($request);
10    return $this->sendLoginResponse($request)
11        ↪ ;
12 }

```

**Kode Sumber IV.6** Implementasi Antarmuka *Login*



### 4.3.3 Melihat Daftar Barang yang Dillang

Halaman ini hanya dapat diakses oleh pengguna yang sudah terdaftar dan sudah *login* ke dalam sistem. Halaman ini menampilkan *form* berisi elemen *input* data diri, dan pengguna dapat mengisi lalu mengklik tombol daftar, dan untuk kasus normal dan alternatif dapat dilihat pada Tabel 3.9.

Terdapat *view logic* khusus pada halaman ini yang ditulis menggunakan Vue dan *dicompile* dengan menggunakan webpack, yang akan dicantumkan dalam Kode Sumber IV.8. Kode sumber implementasi *back-end* dapat dilihat pada Kode Sumber IV.7.

```

1  /*      file : app/Http/Controllers/
      ↳ HomeController*/
2  public function index() {
3      /*      method : GET */
4
5      /*      variabel berisi id barang
6              yang disort dari tanggal
              ↳ perbaruan
7              secara descending */
8      $data['items'] = Item::all()->sortByDesc
      ↳ ('created_at');
9
10     /*      variabel berisi id barang
11             yang sedang aktif proses lelang
12             menggunakan repository :
              ↳ itemRepository */
13     $data['activebid'] = $this->
      ↳ itemRepository->getActiveItem();
14
15
16     return view('pages.general.landing',
      ↳ $data);
17 }

```

---

**Kode Sumber IV.7 Implementasi *Back-end* Melihat Daftar Barang**

```

1  <div>
2    
3
4    <div class="ribbon" v-if="isFavorited"
      ↳ @click.prevent="unFavorite(item)" >
5      <div class="border-ribbon"></div>
6      <i class="fa fa-heart"></i>
7    </div>
8
9    <div class="unribbon" v-else @click.prevent="
      ↳ favorite(item)">
10     <div class="border-ribbon"></div>
11     <i class="fa fa-heart-o"></i>
12   </div>
13 </div>
14
15 export default {
16   props: ['item', 'favorited'],
17   data: function() {
18     return {
19       isFavorited: '',
20       imgUrl : 'http://URL_GAMBAR_DEFAULT'
21     }
22   },
23   mounted() {
24     this.isFavorited = this.isFavorite ? true
25       ↳ : false;
26   },
27   created() {
28     axios.get("get/img/item/" + this.item)
29       .then( response => {
30         if(response.data.replace(/\s+/g, '')
31           ↳ != '' ){
32           var url = /*rewrite image url*/;

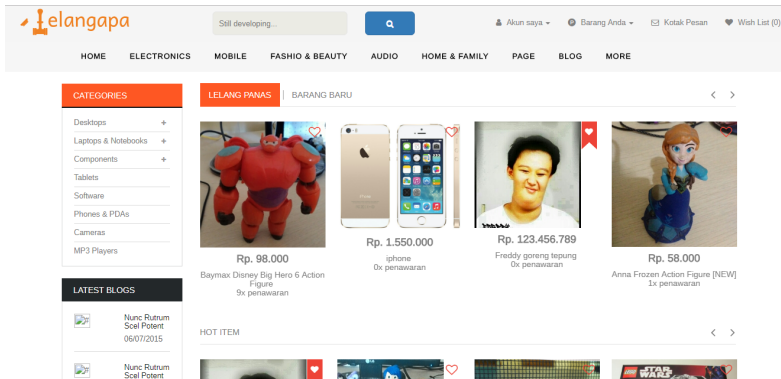
```

```

31         this.imgUrl = url.replace(/\s+/g, '
32             ↪ ');
33         console.log(this.imgUrl);
34     });
35 },
36 computed: {
37     isFavorite()
38     {
39         ↪ return this.favorited;
40     }
41 },
42 methods: {
43     favorite(item) {
44         axios.post('/ajax/favourite/' + item)
45             .then(response =>
46                 ↪ this.isFavorited = true)
47             .catch(response => console.log(
48                 ↪ response.data));
49     },
50     unFavorite(item) {
51         axios.post('/ajax/favourite/un/' + item
52             ↪ )
53             .then(response =>
54                 ↪ this.isFavorited = false)
55             .catch(response => console.log(
56                 ↪ response.data));
57     }
58 }
59 }

```

**Kode Sumber IV.8** Implementasi Vue Melihat Daftar Barang



**Gambar 4.6** Halaman Antarmuka Melihat Barang yang Dilelang

#### 4.3.4 Menawar Barang

Halaman ini hanya dapat diakses oleh pengguna yang sudah terdaftar. Halaman ini menampilkan halaman informasi barang, dan sebuah elemen *input* harga, dan pengguna dapat mengisi lalu mengklik tombol daftar, dan untuk kasus normal dan alternatif dapat dilihat pada Tabel 3.11.

Alur pemrosesan dan logika program pada kasus penggunaan ini dapat dijabarkan sebagai berikut:

1. Pemrosesan *back-end* ditulis menggunakan PHP yang dicantumkan dalam Kode Sumber IV.9;
2. Logika *view* ditulis menggunakan jQuery yang dicantumkan dalam Kode Sumber IV.11; dan
3. Logika proses lelang, berjalan diatas socket yang berjalan diatas Node.js dengan bantuan Socket.io yang dicantumkan dalam Kode Sumber IV.10



## BLENDER 1945

Lorem Ipsum, dolor ipsum set amet.  
 Harga Awal 1000000  
 Awal lelang : 2016-11-30 18:50:00+00  
 Akhir lelang : 2017-07-21 19:50:00+00

---

**PENAWAR TERTINGGI : RAE LARKIN**  
**RP. 3490001**

Tawarkan Harga

Tawar

Description

sdfjaskdfjaskldfjaskldfjasdfafssadf

**Gambar 4.7** Halaman Antarmuka Menawar Barang

```

1  /*      file : app/Http/Controllers/
      ↳ ItemController*/
2  public function show($id) {
3      /*      method : GET */
4
5      /*      tampilkan halaman */
6      $data['item'] = $this->itemRep->find($id);
7
8      if($data['item']->bid_status == 1 )
9      {
10         $data['currentPrice'] = $this->itemRep->
      ↳ getCurrentPrice($id);
11     }
12
13     /*  Jika barang yang disubmit sudah selesai/
      ↳ dimenangkan,
14         tambahkan variabel winner ke view */
15     if($data['item']->bid_status == -1 && $data['

```

```

16         ↪ item']->winner_chosen_status)
        $data['winnername'] = User::find($data['
            ↪ item']->id_user)->name;
17
18     /* tampilkan halaman */
19     return view('pages.'. $this->pageFolder.'.
        ↪ detail', $data);
20 }

```

**Kode Sumber IV.9** Implementasi *Back-end* Menampilkan Halaman Lelang Barang

```

1  /*
2  file : bidserver_https.js
3  menggunakan dependencies : socketio (ioServer),
    ↪ https, http, fs dan express
4  */
5  socket.on('submitbid', function(bidjson) {
6
7      // parameter (bidjson) adalah JSON yang
    ↪ terdiri dari :
8      // id_bidder, id_item, dan harga_bid
9
10     var bidobj = JSON.parse(bidjson);
11     biddingAPI.submitBid(bidobj, function(status,
    ↪ result)
12     {
13         // status adalah return value
14         // jika status = 1, broadcast ke room
15         // untuk mengupdate pemenang lelang saat
    ↪ tersebut.
16         if (status=="1")
17         {
18             //jika sukses
19             var messageObject = {};
20             var tokenArray;
21
22             // fungsi untuk mengconstruct

```

```

23         // return message
24         constructBroadcastMsg(messageObject);
25
26         // broadcast ke semua yang
27         // join ke room lelang tersebut.
28         io.to(result.item_id_return).emit('
           ↳ bidsuccess', result);
29
30     }
31     else if (status=="0"){
32         // jika gagal,
33         // maka send ke sender bahwa bid failed
34         socket.emit('bidfailed', { bidstatus: "
           ↳ failed" });
35     }
36 });
37
38 });

```

**Kode Sumber IV.10** Implementasi Logika Lelang (menggunakan Node.js)

```

1  // script ini berjalan
2  // saat tombol Tawar diklik
3
4  $('.tawar').click(function()
5  {
6      var penawaran = $('#submitted_price').val();
7      var priceNow = $('.priceongoing').html();
8      if(penawaran==""){
9          alertNoPriceSubmitted();
10     }
11     else if(penawaran < priceNow ){
12         alertInvalidBid();
13     }
14     else {
15         var JSONToSend;
16         constructJson(JSONToSend);

```

```

17
18      // submit bid ke socket
19      socket.emit('submitbid', JSONTToSend);
20
21      // tunggu return dari socket
22      // jika hasilnya bid failed, maka
23      ↪ tampilkan pesan error
24      socket.on('bidfailed', function (result)
25      ↪ {
26          alertFailedBid();
27      });
28
29      // jika hasilnya bid sukses,
30      // tampilkan pesan sukses (toast)
31      // dan render ulang halaman (ubah isi
32      ↪ elemen)
33      socket.on('bidsuccess', function (result)
34      ↪ {
35          alertSuccessfulBid();
36          renderPageWithNewPrice(result);
37      });
38  }
39  });

```

**Kode Sumber IV.11** Implementasi Logika UI (menggunakan jQuery)

### 4.3.5 Mendaftarkan Barang untuk Dilelang

Halaman ini hanya dapat diakses oleh pengguna yang sudah terdaftar dan sudah *login* ke dalam sistem. Halaman ini menampilkan *form* berisi elemen *input* informasi barang, dan pengguna dapat mengisi lalu mengklik tombol tambahkan barang, dan untuk kasus normal dan alternatif dapat dilihat pada Tabel 3.13.

Terdapat logika *view* khusus pada implementasi kasus penggunaan ini, karena adanya kebutuhan untuk *upload multiple*



*images* untuk setiap barang dan adanya penggunaan layanan pihak ketiga (AWS Storage Service) sebagai tempat penyimpanan gambar. Kode-kode sumber terkait dengan implementasi kasus penggunaan Mendaftarkan Barang untuk Dilelang adalah sebagai berikut:

1. Logika *back-end* ditulis menggunakan PHP yang dicantumkan dalam Kode Sumber IV.13;
2. Logika *view* ditulis menggunakan jQuery yang dicantumkan dalam Kode Sumber IV.13; dan
3. Logika *upload/storage* data gambar, yang ditulis menggunakan PHP dicantumkan pada Kode Sumber IV.14

Home / Kelola Barang / Tambah Barang

### Jual Barang

\* Nama Barang

\* Deskripsi Barang

\* Foto Barang 

Klik atau seret untuk menambahkan gambar

DETAIL LELANG

\* Waktu Mulai Lelang : July 2017

u Mo Tu We Th Fr Sa

**Gambar 4.8** Halaman Antarmuka Mendaftarkan Barang untuk Dilelang

```

1  /*
2      Fungsi create() dipanggil untuk menampilkan
          ↳ halaman tambah barang
3      Fungsi store() dipanggil saat form di halaman
          ↳ tambah barang diklik
4      File : ItemController
5  */
6  public function create(){
7      /* method : GET */
8      return view('pages.'. $this->pageFolder.'.
          ↳ create', $data);
9  }
10
11
12  /*
13      Fungsi store() dipanggil oleh ajax lewat
          ↳ jQuery
14      untuk memvalidasi dan insert data ke dalam
          ↳ database
15  */
16  public function store(Request $request)
17  {
18      /* method : POST */
19      $unserialize = $this->unserializeForm(
          ↳ $request['data']);
20      /* Validating data */
21      $validate = Validator::make($unserialize,
          ↳ $this->itemRep->rules(), $this->itemRep
          ↳ ->messages());
22
23      if(false){
24          /* Jika gagal, return false dan error
              ↳ message */
25          return ['success'=>false, 'msg' =>
              ↳ $validate->messages()];
26      }
27      else{

```

```

28             /* Jika sukses, return true
29             dan id barang yang berhasil diinsert
30             untuk selanjutnya diproses oleh
                 ↳ browser
31             untuk upload gambar barang ke URL
                 ↳ terpisah.
32             */
33             return ['success' => true, 'id' =>
                 ↳ 10];
34         }
35     }

```

**Kode Sumber IV.12 Implementasi *Back-end* Mendaftarkan Barang untuk Dilelang**

```

1  /*           Fungsi ini berfungsi untuk \textit{update
                 ↳ } gambar barang, diteruskan kepada \textit{
                 ↳ script} terpisah
2           Pada file : ImageController
3  */
4  public function uploadItemImage(Request $request)
                 ↳ {
5           //passed here if csrf token is already passed
6           $_POST['image'] = $request->file('image');
7           $_POST['id_user'] = Auth::user()->id;
8           $_POST['ext'] = $request->file('image')->
                 ↳ extension();
9           require ('/home/img/upload-aws.php');
10          return $res;
11  }

```

**Kode Sumber IV.13 Implementasi *Back-end* Mendaftarkan Barang untuk Dilelang**

```

1  /*           Fungsi ini berfungsi untuk \textit{upload
                 ↳ } gambar lewat script terpisah (untuk
                 ↳ alasan integrasi data dengan sistem yang
                 ↳ dibuat oleh partner tugas akhir)

```

```

2  */
3  require 'vendor/autoload.php';
4
5  use Aws\S3\S3Client;
6
7  $collection = (new MongoDB\Client("mongodb://
    ↪ 127.0.0.1:27017"))->lolangkita->itemimages;
8  if ($_SERVER['REQUEST_METHOD']=='POST'){
9      $image = $_POST['image'];
10     $userid = $_POST['id_user'];
11     $ext = $_POST['ext'];
12     $itemid = $_POST['itemid'];
13     $isMainImage = false;
14
15     if (isset($_POST['is_main_image']) && !
        ↪ empty($_POST['is_main_image']) &&
        ↪ $_POST['is_main_image'] == "true")
        ↪ {
16         $isMainImage = true;
17     }
18
19     $year = date("Y");
20     $month = date("m");
21     $date = date("d");
22     $unique = uniqid();
23     $filename = $userid."_".$unique.".".$ext;
24     $path = "img/".$year."/".$month."/".$date
        ↪ ."/".$userid;
25     $imgpath = $path."/".$filename;
26
27     $decoded_image = base64_decode($image);
28
29     try {
30         $s3client = S3Client::factory(array(
31             'credentials' => array(
32                 'key' => 's3\_key\_credentials',
33                 'secret' => 's3\_secret\_credentials'
34             ),

```

```

35     'profile' => 's3\_profile',
36     'region' => 's3\_selected\_server\_regopm'
37 );
38
39 $upload = $s3client->putObject(
40     array(
41         'Bucket' => 'img-s7.lelangapa.com',
42         'Key' => $imgpath,
43         'Body'    => $decoded_image,
44         'ContentEncoding' => 'base64',
45         'ContentType' => 'image/.'.$ext,
46         'ACL'      => 'public-read'
47     )
48 );
49
50 $insertToDB = $collection->insertOne([
51     'item_id' => $itemid,
52     'url' => $imgpath,
53     'is_main_image' =>
54         ↪ $isMainImage
55 ]);
56
57 if ($isMainImage == true) {
58     $indexImageURL = "https://src-api.
59         ↪ lelangapa.com/apis/index/submit
60         ↪ /image";
61
62     $headers = array(
63         'Content-Type: application/json'
64     );
65     $post_data = array(
66         "id_item" => $itemid,
67         "main_image_url" => $imgpath
68     );
69     $sch = curl_init();
70     curl_setopt($sch, CURLOPT_URL,
71         ↪ $indexImageURL);
72     curl_setopt($sch, CURLOPT_HTTPHEADER,
73         ↪ $headers);

```

```

68         curl_setopt($ch,
           ↪ CURLOPT_RETURNTRANSFER, 1);
69         curl_setopt($ch, CURLOPT_POST, true);
70         curl_setopt($ch, CURLOPT_POSTFIELDS,
           ↪ json_encode($post_data));
71         curl_setopt($ch,
           ↪ CURLOPT_SSL_VERIFYPEER, FALSE);
72         $indexresult = curl_exec($ch);
73         curl_close($ch);
74     }
75
76     if ($upload) {
77         $res = array('status' => 'success', 'result'
           ↪ ' => '1');
78         echo "success";
79     }
80     else {
81         $res = array('status' => 'failed', 'result'
           ↪ => '0');
82         echo "Failed";
83     }
84 }
85 catch(Exception $e) {
86     exit($e->getMessage());
87 }
88 }

```

**Kode Sumber IV.14** Implementasi *Back-end Upload* Gambar Barang

#### 4.3.6 Memperbarui Barang

Halaman ini dapat diakses oleh semua *pengguna*, baik yang belum terdaftar maupun sudah. Halaman ini menampilkan *form* berisi elemen *input* data barang yang sebelumnya sudah terisi data-data barang sebelumnya, dan pengguna dapat memperbarui data lalu mengklik tombol Perbarui, dan untuk kasus alternatif dapat dilihat pada Tabel 3.14.

Pada implementasinya, kasus penggunaan ini kurang lebih sama dengan kasus penggunaan Mendaftarkan Barang (Subbab 4.3.5) karena adanya penggunaan layanan pihak ketiga (AWS Storage Service) sebagai tempat penyimpanan gambar. Kode-kode sumber terkait dengan implementasi kasus penggunaan Mendaftarkan Barang untuk Dilelang adalah sebagai berikut:

1. Logika *back-end* ditulis menggunakan PHP yang dicantumkan dalam Kode Sumber IV.15;
2. Logika *view* ditulis menggunakan jQuery yang dicantumkan dalam Kode Sumber IV.16; dan
3. Logika *update image/storage* data gambar, yang ditulis menggunakan PHP dicantumkan pada Kode Sumber IV.17

```

1  /*
2      Fungsi edit() dipanggil untuk menampilkan
          ↳ halaman perbarui barang
3      Fungsi update() dipanggil saat form di
          ↳ halaman update barang disubmit
4      File : ItemController
5  */
6  public function edit($id)
7  {
8      /*      method : GET */
9      $data['item'] = $this->itemRep->find($id);
10
11     /*  Jika barang sedang aktif lelang,
12         tidak dapat diedit (redirect kembali)
          ↳ */
13     if($data['item']->bid_status==1){
14         return redirect()->back()->with('
          ↳ errorItem','Maaf barang yang sedang
          ↳ dilelang tidak dapat diedit!');
15     }
16     /* otorisasi */
17     if($data['item']->id_user!=Auth::user()->id){

```

```

18         return redirect()->back()->with('
           ↳ errorItem','Maaf anda tidak
           ↳ terotorisasi!');
19     }
20
21     return view('pages.'. $this->pageFolder.'.
           ↳ update', $data);
22 }
23
24
25 /*
26     Fungsi update() dipanggil oleh ajax lewat
           ↳ jQuery
27     untuk memvalidasi dan update data ke dalam
           ↳ database
28 */
29 public function store(Request $request)
30 {
31     /* method : POST */
32     $unserialize = $this->unserializeForm(
           ↳ $request['data']);
33     /* Validating data */
34     $validate = Validator::make($unserialize,
           ↳ $this->itemRep->rules(), $this->itemRep
           ↳ ->messages());
35
36     if(false){
37         /* Jika gagal, return false dan error
           ↳ message
38         ke view
39         */
40         return ['success'=>false, 'msg' =>
           ↳ $validate->messages()];
41     }
42     else{
43         /*
44         Jika sukses, return true
45         dan id barang yang berhasil diinsert

```



```

46         untuk selanjutnya diproses oleh
           ↳ browser
47         untuk upload gambar barang ke URL
           ↳ terpisah.
48     */
49     return ['success' => true, 'id' => 10];
50 }
51 }

```

**Kode Sumber IV.15 Implementasi *Back-end* Memperbarui Barang**

```

1  /* Fungsi ini berfungsi untuk \textit{update}
   ↳ gambar barang, diteruskan kepada \textit{
   ↳ script} terpisah
2  */
3  init: function() {
4      var submitButton = document.querySelector("#
   ↳ click");
5      var myDropzone = this; // closure
6      submitButton.addEventListener("click", function
   ↳ () {
7          swal({
8              text: 'Memperbarui data barang anda...',
9              allowOutsideClick: false,
10             showConfirmButton: false,
11             onOpen: function () {
12                 var dataSubmission = $('#submititem').
   ↳ serialize();
13                 var starttime=
14                 '&start_time=' +
15                 addTimebasedTZ($('#start_time').data().
   ↳ date);
16                 var endtime = '&end_time=' +
   ↳ addTimebasedTZ($('#end_time').data
   ↳ ().date);
17                 dataSubmission += starttime;
18                 dataSubmission += endtime;
19                 $.ajax({

```

```

20     type: "POST",
21     url: '{{ route('item.update') }}',
22     data: { _token: '{{ csrf_token() }}',
           ↪ data : dataSubmission },
23     success: function( msg ) {
24         console.log (msg);
25         if(msg.success == false){
26             $('#.asdf').css('display','
           ↪ block');
27             $('#.asdf').empty();
28             $.each(msg.msg, function(i,
           ↪ val){
29                 $('#.asdf').append(
30                     '<li>'+val+'</li>'
31                 );
32             });
33             swal('Oops',
34                 'Maaf, data anda belum valid.
           ↪ Silahkan cek kembali'
35                 , 'error');
36
37         }
38     else{
39         if(myDropzone.files.length >0)
           ↪ {
40             iditem = msg.id;
41             myDropzone.processQueue();
42         }
43
44     else{
45
46         swal({
47             title: 'Sukses!',
48             type: 'success',
49             allowOutsideClick : false
           ↪ ,
50             showConfirmButton : false
           ↪ ,

```

```

51         text: "Sukses menambahkan
52             ↳ barang!",
53         timer: 1000
54     }).then(function () {
55         document.location = "{{
56             ↳ url('item/') }}" +
57             ↳ iditem;
58     });
59 }
60 }
61 });
62 });
63
64 this.on("processing", function() {
65     swal('Uploading image..');
66 });
67
68 this.on("sending", function(file, xhr, data) {
69     if(iditem != 0){
70         data.append("_token", "{{ csrf_token()
71             ↳ }}");
72         data.append("itemid", iditem);
73     }
74 });
75 },
76 /*
77     saat gagal error gambar,
78     bagian fungsi error dipanggil
79 */
80 error: function(file, response) {
81     swal({
82         title: 'Oops',
83         type: 'error',
84         allowOutsideClick: false,
85         showConfirmButton: false,

```

```

85         text: "Maaf ada kesalahan upload gambar,
           ↳ silahkan upload gambar di halaman
           ↳ edit gambar.",
86         timer: 2000
87     }).then(function () {
88         document.location = "{{ url('item') }}"
           ↳ + iditem;
89     });
90
91 },
92 /*
93     saat sukses update gambar,
94     bagian fungsi error dipanggil
95 */
96 success: function(file,done) {
97     swal({
98         title: 'Sukses!',
99         allowOutsideClick: false,
100         showConfirmButton:false,
101         type:'success',
102         text: "Sukses memperbarui barang!",
103         timer: 1000
104     }).then(function () {
105         document.location = "{{ url('item/') }}"
           ↳ + iditem;
106     });
107 }
108
109 });

```

**Kode Sumber IV.16** Implementasi *View* Memperbarui Barang

```

1  /*      Fungsi ini berfungsi untuk \textit{upload
           ↳ } gambar lewat script terpisah (untuk
           ↳ alasan integrasi data dengan sistem yang
           ↳ dibuat oleh partner tugas akhir)
2
3  */

```

```

4 use Aws\S3\S3Client;
5
6 $collection = (new MongoDB\Client("mongodb://
    ↪ 127.0.0.1:27017"))->lelangita->itemimages;
7 if ($_SERVER['REQUEST_METHOD']=='POST'){
8     constructFilePath();
9     $imgpath = $path."/". $filename;
10
11     $decoded_image = base64_decode($image);
12
13     try {
14         $s3client = S3Client::factory(array(
15             'credentials' => array(
16                 'key' => 's3\_key\_credentials',
17                 'secret' => 's3\_secret\_credentials'
18             ),
19             'profile' => 's3\_profile',
20             'region' => 's3\_selected\_server\_region'
21         ));
22
23         $upload = $s3client->putObject(
24             array(
25                 'Bucket' => 'img-s7.lelangapa.com',
26                 'Key' => $imgpath,
27                 'Body' => $decoded_image,
28                 'ContentEncoding' => 'base64',
29                 'ContentType' => 'image/'. $ext,
30                 'ACL' => 'public-read'
31             )
32         );
33
34         $insertToDB = $collection->insertOne([
35             'item_id' => $itemid,
36             'url' => $imgpath,
37             'is_main_image' =>
    ↪ $isMainImage
38         ]);
39

```

```

40     if ($isMainImage == true) {
41         $indexImageUrl = "https://src-api.
            ↪ lelangapa.com/apis/index/submit/
            ↪ image";
42         $headers = array(
43             'Content-Type: application/json'
44         );
45         $post_data = array(
46             "id_item" => $itemid,
47             "main_image_url" => $imgpath
48         );
49         $ch = curl_init();
50         curl_setopt($ch, CURLOPT_URL,
            ↪ $indexImageUrl);
51         curl_setopt($ch, CURLOPT_HTTPHEADER,
            ↪ $headers);
52         curl_setopt($ch, CURLOPT_RETURNTRANSFER,
            ↪ 1);
53         curl_setopt($ch, CURLOPT_POST, true);
54         curl_setopt($ch, CURLOPT_POSTFIELDS,
            ↪ json_encode($post_data));
55         curl_setopt($ch, CURLOPT_SSL_VERIFYPEER,
            ↪ FALSE);
56         $indexresult = curl_exec($ch);
57         curl_close($ch);
58     }
59 }
60 else {
61     $collection->updateOne(
62         ['_id' => new \MongoDB\BSON\ObjectID(
            ↪ $imageid)],
63         ['$set' => array('url' => $imgpath, '
            ↪ is_main_image' => $isMainImage)]
64     );
65 }
66
67 if ($upload) {
68     $res = array('status' => 'success', 'result

```

```

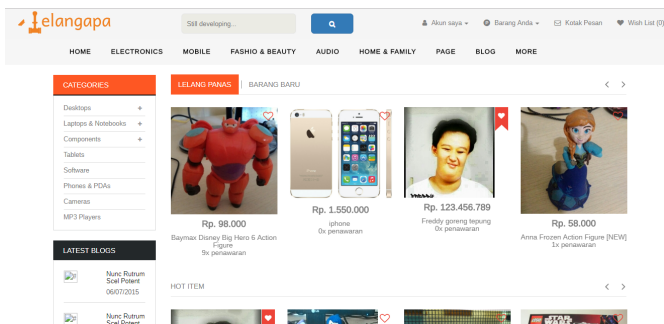
69         => '1');
70         echo "success";
71     }
72     else {
73         $res = array('status' => 'failed', 'result'
74             => '0');
75         echo "Failed";
76     }
77 }
78 catch(Exception $e) {
79     exit($e->getMessage());
80 }
81 }

```

**Kode Sumber IV.17** Implementasi *Back-end Upload* Gambar Barang

### 4.3.7 Melihat Barang yang Didaftarkan

Halaman ini hanya dapat diakses oleh pengguna yang sudah terdaftar dan sudah *login* ke dalam sistem. Halaman ini berisi daftar barang yang pernah didaftarkan pengguna, sesuai dengan spesifikasi kasus penggunaan 3.15. Tidak ada *view logic* ataupun logika UI khusus dalam halaman ini. Kode sumber implementasi *back-end* dapat dilihat pada Kode Sumber IV.18.



**Gambar 4.9** Halaman Antarmuka Melihat Barang yang Pernah Didaftarkan

```

1  /*
2      file : app/Http/Controllers/
          ↳ ItemController
3      terkait dengan ItemRepository
4  */
5  public function index(){
6      /*      method : GET */
7
8      $data['items'] = $this->itemRep->getUserItem
          ↳ ();
9      return view('pages.'. $this->pageFolder.'.
          ↳ index', $data);
10 }
11
12 /*      file : app/Http/Controllers/
          ↳ ItemRepository */
13 public function getUserItem()
14 {
15     return Auth::user()->item()->orderBy('
          ↳ end_time', 'desc')->get();
16 }

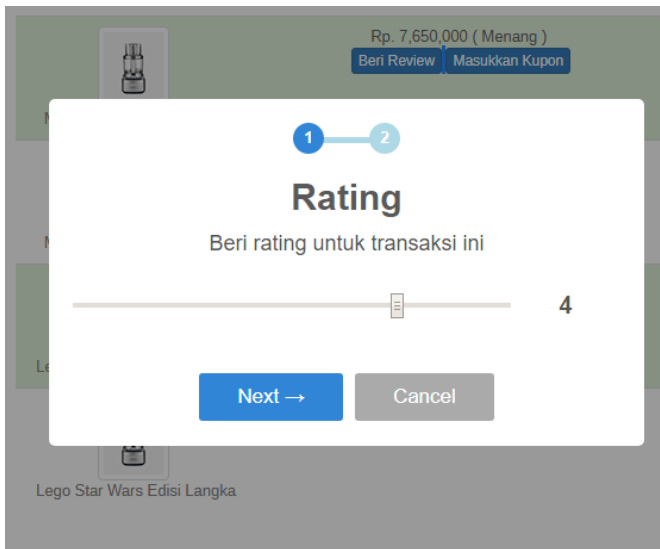
```

**Kode Sumber IV.18** Implementasi *Back-end* Melihat Barang yang Pernah Didaftarkan

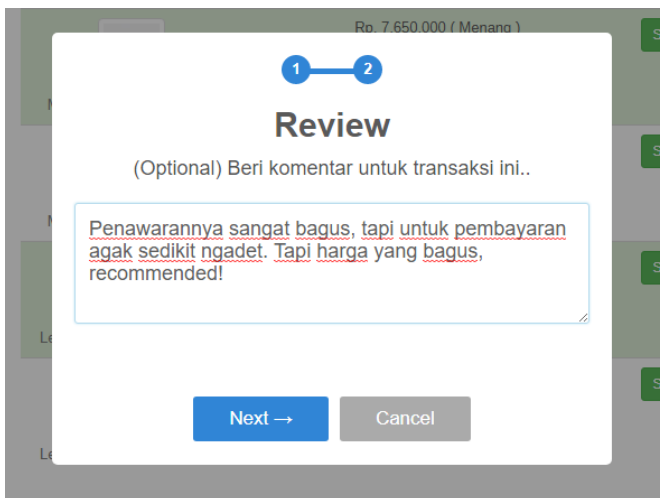
#### 4.3.8 Menambahkan Review

Halaman ini hanya dapat diakses oleh pengguna terdaftar yang sudah *login*, dan hanya bisa menambahkan review kepada transaksi yang sudah selesai. Halaman ini menampilkan form *modal* berisi *input* data *review*, memberikan *rating*. Pengguna dapat mengisi lalu mengklik tombol daftar, untuk kasus alternatif dapat dilihat pada Tabel 3.18. Logika *back-end* ditulis menggunakan PHP yang dicantumkan dalam Kode Sumber IV.19;





**Gambar 4.10** Halaman Antarmuka Implementasi Menambahkan *Review*



**Gambar 4.11** Halaman Antarmuka Implementasi Menambahkan *Review*

```

1  /*      file : app/Http/Controllers/
    ↪ ReviewController */
2  /* fungsi ini dieksekusi saat review disubmit */
3
4  public function submit(Request $request, $idbid)
5  {
6      $informationNeeded = json_decode($request['
    ↪ info']);
7
8      $bid = BidRepository::getBidDetail($idbid);
9      $user = $request->user();
10     if( !$bid || ($user->id != $bid->id_user &&
    ↪ $user->id!= $bid->item->user->id) )
11         abort(403);
12
13     return response()
14         ->json(
15         ReviewRepository::submit($bid,
    ↪ $informationNeeded, $user)
16     );
17 }
18
19 /*      file : app/Http/Controllers/
    ↪ ReviewRepository */
20 public static function submit(Bid $bid,
    ↪ $information, User $user)
21 {
22     $constructArray = [
23         'rate' => $information[0],
24         'rate_message' => $information[1]
25     ];
26
27     $validator = Validator::make($constructArray,
    ↪ self::rule(), self::message());
28
29     if ($validator->fails()) {
30         return ['valid'=>false, 'msg'=>

```

```

31         ↪ $validator->errors() ];
32     }
33     //stting user yang mau dirate as $user2
34     if($user->id == $bid->id_user)
35     {
36         $user2 = $bid->item->user;
37         $isBidder = true;
38     }
39     else if($user->id == $bid->item->user->id){
40         $user2 = $bid->user;
41         $isBidder = false;
42     }
43
44     $constructQuery = [
45         'id_user_rater' => $user->id,
46         'nama_user_rater' => $user->name,
47         'rate' => $constructArray['rate'],
48         'rate_message' => $constructArray['
49             ↪ rate_message'],
50         'id_item' => $bid->id_item,
51         'bid_time' => $bid->bid_time
52     ];
53     //         merge with
54
55     if($isBidder)
56         array_merge($constructQuery,
57             ['id_user_auctioneer'=> $user2->id, '
58                 ↪ nama_user_auctioneer' => $user2
59                 ↪ ->name]);
60
61     else
62         array_merge($constructQuery,
63             ['id_user_bidder'=> $user2->id, '
64                 ↪ nama_user_bidder' => $user2->
65                 ↪ name]);
66
67     $id = DB::table( $isBidder ? '

```

```

63         ↪ ratingauctioneers' : 'ratingbidders')
        ↪ ->insertGetId($constructQuery);
64
65     if($id) return ['valid' => true];
66     else return ['valid' => false, 'msg' => '
        ↪ Maaf kesalahan server, silahkan
        ↪ coba lagi'];
67
68     }
69
70     /*         file : app/Http/Controllers/BidRepository
        ↪         */
71     public static function getBidDetail($idbid){
72         return Bid::findOrFail($idbid);
73     }

```

**Kode Sumber IV.19** Implementasi *Back-end* Menambahkan Review

### 4.3.9 Mengirimkan Pesan

Halaman ini hanya dapat diakses oleh pengguna yang sudah terdaftar dan masuk/*login* ke dalam sistem. Pada halaman ini terdapat elemen-elemen halaman *chatting* pada umumnya, yaitu elemen *input* pesan, tombol Kirim, dan riwayat beberapa pesan sebelumnya. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.19.

Terdapat logika *view* dan alur proses khusus dikarenakan sifat pengiriman dan penerimaan pesan yang realtime, sehingga dibangun diatas Node.js dan menggunakan Socket.io. Masing-masing logika tersebut dapat dijabarkan sebagai berikut:

1. Logika *back-end* ditulis menggunakan PHP yang dicantumkan dalam Kode Sumber IV.20;
2. Logika *view* ditulis menggunakan jQuery yang dicantumkan dalam Kode Sumber IV.22; dan
3. Logika proses pengiriman & penerimaan pesan, berjalan diatas *runtime* Node.js dengan bantuan Socket.io yang dicantumkan dalam Kode Sumber IV.21



**Gambar 4.12** Halaman Antarmuka Kasus Penggunaan Mengirimkan Pesan

```

1  /*      file : app/Http/Controllers/
      ↳ ChatController */
2  public function chat($id_user)
3      /*      method : GET */
4
5      if(intval($id)){
6          $data['from'] = Auth::user()->id;
7          $data['to'] = $id;
8          $data['user'] = User::findOrFail($id);
9
10         if($data['user']){
11             return view('pages.user.chat', $data)
12                 ↳ ;
13         }
14
15         /*  Jika ada parameter error,
16             redirect kembali dengan pesan error */
17         return redirect('/')->with('error','User yang
18             ↳ anda cari tidak dapat ditemukan.');
```

**Kode Sumber IV.20 Implementasi *Back-end* Mengirimkan Pesan**

```

1  /*
2  file : chatserver\_https.js
3  menggunakan dependencies : socketio (ioServer),
4      ↳ jwt, https, http, fs dan express
5  */
6  socket.on('join-room', function(room) {
7      /*
8      fungsi ini dipanggil saat
9      pertama kali pengguna membuka halaman chat
10     dimana pengguna masuk ke dalam room percakapan
11     */
12     var parsedRoom = parseRoom(room);
```

```

12
13
14  /*
15     Jika room ini belum pernah dimasuki/ chat
16     ↪ pertama
17     maka ditambahkan ke tabel joinedroom
18  */
19     insertToRoomCollection(roomcollection,
20     ↪ parsedRoom, false, function() {
21         socket.join(parsedRoom.room);
22
23         //
24         var cb = function(err, chat) {
25             if (chat!==[]){
26                 socket.emit('chathistory', chat);
27             }
28         };
29
30     /*
31     Broadcast 5 percakapan terakhir
32     dalam room tersebut
33     ke pengguna yang baru saja join room.
34  */
35     collection.find({ room : parsedRoom.room
36     ↪ }).sort({ sent : -1 }).limit(50).
37     ↪ toArray(cb);
38
39     });
40
41 });
42
43 socket.on('send', function(data) {
44     /*
45     parameter yang masuk adalah JSON dengan
46     ↪ konstruksi berikut :
47     { room: , body : }
48  */
49

```

```

45     var msgParse = JSON.parse(data);
46     var parsedRoom = parseRoom(msgParse.room);
47
48     //constructing inserting query
49     var insertQuery = {
50         room : parsedRoom.room,
51         sender : parsedRoom.from,
52         msg : msgParse.body,
53         sent : new Date()
54     };
55
56     //insert kedalam database
57     collection.insert(insertQuery, function(err,
58         ↪ o) {
59         var ll = parsedRoom.from.toString();
60         if (err) io.to(ll).emit('send-status', {
61             ↪ status : false});
62         else io.to(ll).emit('send-status', {
63             ↪ status : true });
64     });
65
66     // update room information
67     // bahwa room ini sudah diperbarui
68     insertToRoomCollection(roomcollection,
69         ↪ parsedRoom, true);
70
71     //broadcast ke user yang tergabung ke room
72     ↪ percakapan
73     io.to(parsedRoom.room).emit('new-msg',
74         ↪ insertQuery);
75
76     });

```

**Kode Sumber IV.21** Implementasi Logika *View* Lelang (menggunakan Node.js)

```

1  var room = '{ { Auth::user()->id } }-{ { $user->id
    ↪ } }';

```



```

2  var prepstat = '{ "room" : ' + room + ', "body"
    ↪      : "';
3  var closetag = '"}';
4
5  /*
6   Masuk ke dalam room
7   */
8  socket.emit('join-room', room);
9
10
11 /*
12  Menerima riwayat perpesanan dan merender
    ↪ pesan-pesan tersebut ke view dengan fungsi
    ↪ bantu appendNewChat()
13  */
14 socket.on('chathistory', function(data) {
15     $(data).each(function(index,value) {
16         appendNewChat(value);
17     });
18 });
19
20 $('.sendMessage').click(function() {
21     /*
22     constructing JSON message
23     concat previous statement with message's body
24     and close tag
25     */
26     var msg = prepstat + $('.body').val() +
        ↪ closetag ;
27
28     /* disable input pesan untuk sementara */
29     $(".body").attr("disabled", true);
30
31     /*kirirkan pesan */
32     socket.emit('send', msg );
33
34     /* menunggu status pengiriman pesan */
35     socket.on('send-status', function(data) {

```

```

36      /* Jika gagal, enable input pesan dengan
      ↳ tidak menghapus pesan yang belum jadi
      ↳ dikirimkan sebelumnya */
37      if(!data.status){
38          swal('Oops','Pesan anda tidak dapat
      ↳ terkirim, silahkan coba lagi','
      ↳ error');
39      }
40      /* Jika sukses, enable input pesan dengan
      ↳ elemen input pesan yang sudah
      ↳ dikosongkan */
41      else if(data.status){
42          $('#body').val('');
43      }
44
45      }, function(){
46          $(".body").attr("disabled", false);
47      });
48  });
49
50  /*
51  Bagian ini dieksekusi saat ada pesan baru masuk
      ↳ ke dalam room. appendNewChat() adalah
      ↳ fungsi bantu merender view untuk
      ↳ menampilkan pesan baru
52  */
53  socket.on('new-msg', function (value) {
54      appendNewChat(value);
55  });

```

**Kode Sumber IV.22** Implementasi Logika Pengiriman & Penerimaan  
Pesan (menggunakan jQuery)












#### 4.3.10 Melihat Kotak Pesan

Halaman ini hanya dapat diakses oleh pengguna yang sudah terdaftar dan masuk/*login* ke dalam sistem. Pada halaman ini terdapat elemen-elemen halaman *chatting* pada umumnya, yaitu elemen *input* pesan, tombol Kirim, dan riwayat beberapa pesan sebelumnya. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.20.

Terdapat logika *view* dan alur proses khusus dikarenakan sifat pengiriman dan penerimaan pesan yang *realtime*, sehingga dibangun diatas Node.js dan menggunakan Socket.io. Masing-masing logika tersebut dapat dijabarkan sebagai berikut:

1. Logika *back-end* ditulis menggunakan PHP yang dicantumkan dalam Kode Sumber IV.23; dan
2. Logika *view* ditulis menggunakan Vue.js yang dicantumkan dalam Kode Sumber IV.24;

##### Kotak Pesan Anda

	Lucinda Pours test	4 days ago 
	Romario Wijaya lol	6 days ago 
	Bebek Surga jangan menyebarkan ke goblokan	1 month ago 
	Freddy Hermawan koplak	1 month ago
	Alexandrea Gleichner Halo brader	1 month ago 
	asem saja test	1 month ago 

**Gambar 4.13** Halaman Antarmuka Melihat Kotak Pesan

```

1  /*      file : app/Http/Controllers/
    ↳ InboxController */
2  /* fungsi ini dipanggil saat mengakses url inbox/
    ↳ data */
3  public function retrieveInbox($id = false)
4  {
5      $r = $this->chatRep->conversations(
6          $id ?
7          User::findOrFail($id) :
8          Auth::user()
9      );
10
11      return response()
12          ->json($r);
13  }
14
15  /* file : ChatRepository */
16  /* fungsi ini dipanggil saat mengakses url inbox/
    ↳ lastDetail */
17  public function conversations(User $user)
18  {
19      $this->chatroom->setIdComp($user->id);
20      return $this->chatroom
21          ->where('id_user_1',
22              ↳ strval($user->id))
23          ->orWhere('id_user_2', strval($user->
24              ↳ id))
25          ->with('friend')
26          ->orderBy('updatedAt', 'desc')
27          ->get();
28  }
29
30  public static function getLastChatDetail(User
31      ↳ $user, $room)
32  {
33      $res = DB::connection('mongodb')
34          ->collection('userchat')

```



```

11         <strong>{{ conv.friend.name }}</
           ↪ strong>
12     </div>
13     <div class="last-message text-muted"
           ↪ >
14         <!--i want to show last chat
           ↪ here-->
15         {{ det[index].msg }}
16     </div>
17     <small class="time text-muted">
18         <!-- i want to show the msg
           ↪ timestamp here-->
19         {{ det[index].sent }}
20     </small>
21     <small class="chat-alert text-muted"
           ↪ >
22         <i v-if="det[index].isSender"
           ↪ class="fa fa-reply"></i>
23     </small>
24     </div>
25 </div>
26 </a>
27 </li>
28 </ul>
29 <ul class="friend-list" v-else>
30     <li>
31         <div class="friend-name">
32             No messages!
33         </div>
34     </li>
35 </ul>w
36 </template>
37
38 <script>
39     export default
40     {
41         props: ['id'],
42         data: function () {

```

```

43     return {
44         conversations: [],
45         det : [],
46         loaded : false
47     }
48 },
49 created() {
50     axios.get('/inbox/data')
51     .then(response => {
52         //fetching conversations id and its
53         ↪ detail
54
55         this.conversations = response.data;
56
57         var promises = [];
58
59         response.data.forEach(conv => {
60             promises.push(
61                 axios.post('/inbox/lastDetail',{
62                     room : conv.room
63                 })
64             );
65
66             axios.all(promises).then( results => {
67                 this.det = results.map(result =>
68                     ↪ result.data);
69                 this.loaded = true;
70             })
71         });
72     },
73     methods: {
74         getUrl(id) {
75             return ("https://lelangapa.com/chat/" + id
76                 ↪ );
77         }
78     }
79 }

```

78 `</script>`

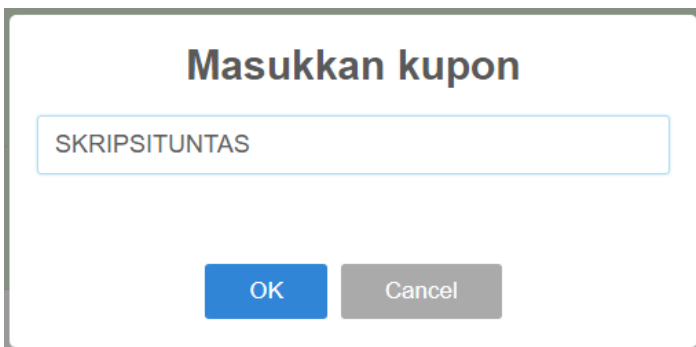
**Kode Sumber IV.24** Implementasi Logika *View* Lihat Kotak Pesan  
(menggunakan Node.js)

#### 4.3.11 Memasukkan Kupon

Halaman ini hanya dapat diakses oleh pengguna yang sudah terdaftar dan masuk/*login* ke dalam sistem. Pada halaman ini terdapat elemen-elemen halaman *chatting* pada umumnya, yaitu elemen *input* pesan, tombol Kirim, dan riwayat beberapa pesan sebelumnya. Spesifikasi kasus penggunaan dapat dilihat pada Tabel 3.20.

Terdapat logika *view* dan alur proses khusus dikarenakan sifat pengiriman dan penerimaan pesan yang realtime, sehingga dibangun diatas Node.js dan menggunakan Socket.io. Masing-masing logika tersebut dapat dijabarkan sebagai berikut:

1. Logika *back-end* ditulis menggunakan PHP yang dicantumkan dalam Kode Sumber IV.25; dan
2. Logika *view* ditulis menggunakan jQuery yang dicantumkan dalam Kode Sumber IV.26;



The image shows a web interface for entering a coupon. At the top, the title "Masukkan kupon" is displayed in a bold, dark font. Below the title is a text input field with a light blue border, containing the text "SKRIPSITUNTAS". At the bottom of the form, there are two buttons: a blue button labeled "OK" and a gray button labeled "Cancel". The entire form is enclosed in a thin gray border.

**Gambar 4.14** Halaman Antarmuka Memasukkan Kupon



```

1  /**
2  * File : app/Http/Controllers/CouponController
3  * Memproses request ajax
4  * Method : any
5  */
6  public function submitCoupon(Request $request)
7  {
8      $id_bid = $request['bid'];
9      $bid = BidRepository::getBidDetail(
10         ↪ $id_bid);
11      $user = Auth::user();
12      $result = $this->couponRepository
13         ->submitCoupon(
14         $user
15         , $bid
16         , $request['coupon']
17     );
18     return response()
19         ->json($result);
20 }

```

**Kode Sumber IV.25** Implementasi *Back-end* Memasukkan Kupon

```

1  /**
2  * Menggunakan tools SweetAlert
3  */
4  swal({
5      title: 'Masukkan kupon',
6      input: 'text',
7      showCancelButton: true,
8      inputValue: function (value) {
9          return new Promise(function (resolve, reject) {
10             ↪ ) {
11                 if (value) {
12                     resolve()
13                 } else {
14                     reject('Kupon tidak boleh kosong!')
15                 }
16             }
17         })
18     }
19 })

```

```

14         }
15     })
16 }
17 }).then(function (couponName) {
18     $.ajax({
19         url : "{{ route('coupon.submit') }}",
20         type : 'post',
21         data: { token : token,
22                 _token : "{{ csrf_token() }}",
23                 bid : bid ,
24                 coupon: "couponName" },
25         xhrFields: {
26             withCredentials: true
27         }
28     })
29     .fail(function (jqXHR, textStatus, errorThrown
30         ↪ ) {
31         return new Promise(function (resolve, reject
32             ↪ ) {
33             if (value) {
34                 resolve()
35             } else {
36                 reject(errorThrown)
37             }
38         });
39     })
40     .done(function (result, textStatus, jqXHR) {
41         var resp = JSON.parse(result);
42         if(resp.status){
43             var msgShow;
44             if(resp.new_price == -1) msgShow = 'Anda
45                 ↪ mendapat promo ongkos kirim';
46             else if (resp.new_price>0) msgShow = '
47                 ↪ Harga barang menjadi ' +
48                 ↪ result.new_price;
49             msgShow = msgShow + "Anda akan
50                 ↪ dihubungi admin untuk lebih

```

```

    ↪ lanjutnya :)");
46     swal (
47         'Sukses!',
48         msgShow,
49         'success'
50     );
51 }
52 else showStatus(resp.msg, false);
53 });
54 })
```

**Kode Sumber IV.26** Implementasi *View Logic* Memasukkan Kupon

4.3.12 Melihat Daftar Pengguna

Halaman ini hanya dapat diakses oleh *administrator* yang sebelumnya sudah *login*. Tidak ada *view logic* ataupun logika *UI* khusus dalam halaman ini. Kode sumber implementasi *back-end* dapat dilihat pada Kode Sumber IV.27.

Pengguna				
Daftar Pengguna				
#	Nama	Email	Status Verifikasi	Tanggal Verifikasi
1	Lauryn Romaguera	deja19@example.net	<a href="#">Belum Terverifikasi</a>	-
2	Elmore Christiansen	amoore@example.com	<a href="#">Belum Terverifikasi</a>	-
3	Michelle Ratke DDS	vhoppe@example.net	<a href="#">Belum Terverifikasi</a>	-
4	Otilia Kilback	srunolfsson@example.com	<a href="#">Belum Terverifikasi</a>	-
5	Dr. Sven Bogan Jr.	kpacochoa@example.net	<a href="#">Belum Terverifikasi</a>	-
6	Una Streich	okuneva.emmett@example.org	<a href="#">Belum Terverifikasi</a>	-
7	Lucinda Pouros	jaids91@example.net	<a href="#">Belum Terverifikasi</a>	-
8	Mrs. Florence Cassin DVM	kirlin.myriam@example.net	<a href="#">Belum Terverifikasi</a>	-

**Gambar 4.15** Halaman Antarmuka Kasus Penggunaan Melihat Daftar Pengguna

```

1  /**
2   * File : app/Http/Controllers/UserController
3   * Menampilkan halaman daftar pengguna
4   * langsung diFetch dari base Model User
5   * Method : GET
6   */
7  public function user()
8  {
9      $data['data'] = User::paginate(20);
10     return view('pages.user.index', $data);
11 }

```

**Kode Sumber IV.27** Implementasi Antarmuka Melihat Daftar Pengguna

#### 4.3.13 Menambahkan Kupon

Halaman ini hanya dapat diakses oleh *administrator* yang sebelumnya sudah *login*. Halaman ini menampilkan form berisi elemen *input* informasi kupon, dan setelah selesai lalu mengklik tombol Tambahkan Kupon, dan untuk kasus alternatif dapat dilihat pada Tabel 3.21.

Tidak ada *view logic* ataupun logika *UI* khusus dalam halaman ini. Kode sumber implementasi *back-end* dapat dilihat pada Kode Sumber IV.28.

```

1
2  /** File : app/Http/Controllers/CouponController
3   * Menampilkan halaman tambah kupon
4   * Method : GET
5   */
6  public function create()
7  {
8      $data['user'] = $this->userRepository->
          ↪ checkbox();
9      return view('pages.coupon.create',$data);
10 }
11
12 /**
13  * Store data kupon yang dimasukkan
14  * fungsi ini dijalankan ketika form disubmit
15  * terkait dengan CouponRepository
16  */
17
18 public function store(Request $request)
19 {
20     $data = $request->all();
21     $data['limit_usages'] = $data['usages'];
22     $ss = $this->couponRepository->storeCoupon(
          ↪ $data);
23
24     if($ss) return redirect('coupon')->with('
          ↪ success',true);
25     else return redirect('coupon')->with('success
          ↪ ',false);
26 }
27 /**
28  * File CouponRepository
29  */
30 public function storeCoupon($data)
31 {
32     /* Menggunakan base model Laravel */
33     /* return status store data ke dalam

```

```

34         ↪ database */
35     return Coupon::create($data);
36 }

```

**Kode Sumber IV.28** Implementasi Antarmuka Menambahkan Kupon

**Tambah Kupon**

**Teks Kupon**

HAPPYBIDDINGJUNE

**Waktu Penggunaan**

**Batas Awal:**

07/06/2017

**Batas Akhir:**

07/13/2017

**Teks Ketentuan**

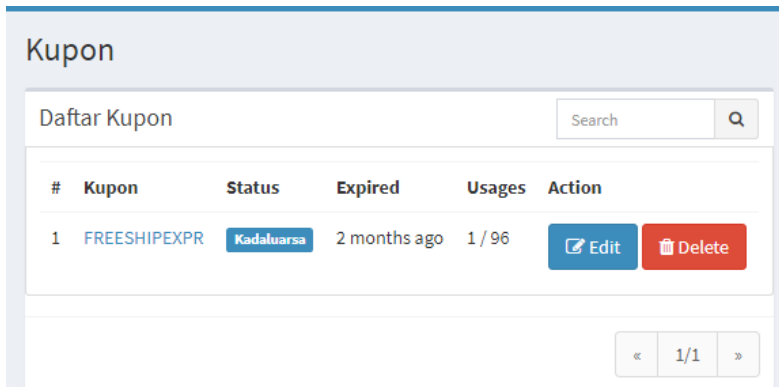
Promo ini memberikan diskon 2.5% bagi anda yang berbelanja sebelum tanggal 13 Juli 2017. Enjoy this coupon!

**Gambar 4.16** Halaman Antarmuka Kasus Penggunaan Menambahkan Kupon

#### 4.3.14 Melihat Daftar Kupon

Halaman ini hanya dapat diakses oleh *administrator* yang sebelumnya sudah *login*. Detail kasus penggunaan dapat dilihat Tabel 3.23.

Tidak ada *view logic* ataupun logika *UI* khusus dalam halaman ini. Kode sumber implementasi *back-end* dapat dilihat pada Kode Sumber IV.29.



**Gambar 4.17** Halaman Antarmuka Kasus Penggunaan Melihat Daftar Kupon

```

1
2  /** File : app/Http/Controllers/CouponController
3   * Menampilkan halaman daftar kupon
4   * tidak terkait dengan CouponRepository
5   * langsung diFetch dari base Model Coupon
6   * Method : GET
7   */
8
9  public function index()
10 {
11     $data['data'] = Coupon::orderBy('activedate',
12         ↪ 'desc')->paginate(20);
13     return view('pages.coupon.index', $data);
14 }

```

**Kode Sumber IV.29** Implementasi Antarmuka Melihat Daftar Kupon

### 4.3.15 Memperbarui Kupon

Halaman ini hanya dapat diakses oleh *administrator* yang sebelumnya sudah *login*. Detail kasus penggunaan dapat dilihat

Tabel 3.23.

Tidak ada *view logic* ataupun logika *UI* khusus dalam halaman ini. Kode sumber implementasi *back-end* dapat dilihat pada Kode Sumber IV.30.

```

1
2  /** File : app/Http/Controllers/CouponController
3   * Menampilkan halaman perbarui kupon
4   * Method : GET
5   */
6  public function edit($id)
7  {
8      try{
9          $data['user'] = $this->userRepository->
            ↪ checkbox();
10         $data['coupon'] = $this->couponRepository
            ↪ ->find($id);
11         return view('pages.coupon.edit', $data);
12     }
13     catch (\Exception $e){
14         abort(403);
15     }
16 }
17 /**
18  * Update data kupon yang dimasukkan
19  * fungsi ini dijalankan ketika form di halaman
            ↪ edit kupon disubmit
20  * terkait dengan CouponRepository
21  */
22
23 public function update(Request $request, $id)
24 {
25
26     $data = $request->all();
27     if($this->couponRepository->update($id, $data
            ↪ )){
28         return redirect('coupon')->with('success

```



```

29         ↪ ', 'Sukses memperbarui kupon!');
30     }
31     else return redirect('coupon')->with('error
32         ↪ ', 'Mohon maaf, gagal memperbarui kupon.
33         ↪ Silahkan coba lagi');
34 }
35 /**
36  * File CouponRepository
37  */
38 public function update($id, $data)
39 {
40     $coupon = $this->find($id);
41     return $coupon->update($data);
42 }

```

**Kode Sumber IV.30** Implementasi *Back-end* Kasus Penggunaan  
Memperbarui kupon

*(Halaman ini sengaja dikosongkan)*

## **BAB V**

### **PENGUJIAN DAN EVALUASI**

#### **5.1 Pengujian**

Aplikasi yang telah dibangun dapat dikatakan berhasil jika sudah memenuhi kebutuhan fungsional dan nonfungsional yang telah didefinisikan pada Tabel 3.1 dan 3.2. Pada bab ini akan dibahas mengenai pengujian dan evaluasi pada aplikasi yang dikembangkan. Pengujian yang dilakukan terdiri dari dua pengujian yaitu pengujian fungsionalitas sistem dan pengujian statistik. Pengujian fungsionalitas mengacu pada kebutuhan fungsionalitas yang didefinisikan pada Tabel 3.1. Sedangkan pengujian statistik dilakukan untuk menguji apakah aplikasi telah memenuhi kebutuhan nonfungsional yang telah didefinisikan pada Tabel 3.2.

##### **5.1.1 Metode Pengujian**

Metode pengujian dirancang sesuai dengan kebutuhan fungsional dan nonfungsional yang telah didefinisikan. Metode-metode pengujian yang digunakan pada tugas akhir ini adalah sebagai berikut:

- 1. Pengujian Fungsionalitas**

Pengujian fungsionalitas sistem dilakukan secara mandiri dengan menyiapkan sejumlah skenario. Deskripsi proses pengujian secara lengkap akan dijelaskan pada Subbab 5.1.2.

- 2. Pengujian Kecepatan**

Pengujian kecepatan dilakukan pada setiap kasus penggunaan, yaitu pencatatan waktu yang untuk menampilkan sebuah halaman dan atau melakukan sebuah *request* ke aplikasi. Deskripsi proses pengujian performa akan secara lengkap dijelaskan pada Subbab 5.1.3.

- 3. Pengujian User Experience**

Pengujian *user experience* dilakukan secara statistik, untuk

menguji apakah bahwa benar aplikasi yang dibangun memberikan *positive user experience* kepada penggunanya. *Key performance indicator* yang digunakan didasarkan pada paper “*Development of an Instrument Measuring User Satisfaction of the Human-Computer Interface*”. Deskripsi proses pengujian performa akan secara lengkap dijelaskan pada Subbab 5.1.4.

#### 4. Pengujian *Maintainability*

Pengujian *maintainability* dimaksudkan untuk menguji apakah benar aplikasi yang dibangun bersifat *maintainable* kepada *developer*. Pengujian ini dilakukan secara statistik, dengan mengacu kepada paper “*A Software Maintainability Evaluation Methodology*”. Deskripsi proses pengujian secara lengkap akan dijelaskan pada Subbab 5.1.5.

### 5.1.2 Pengujian Fungsionalitas

#### 5.1.2.1 Pengujian Fungsionalitas Manajemen Autentikasi

**Tabel 5.1** Pengujian Fungsionalitas Fitur Manajemen Akun

<b>ID</b>		<b>UJ-1-KP01</b>
<b>Referensi</b>	<b>Kasus</b>	KP01
<b>Penggunaan</b>		
<b>Nama</b>		Pengujian fitur manajemen autentikasi
<b>Skenario 1</b>		Menguji fitur registrasi
Kondisi Awal		Sistem menampilkan halaman registrasi beserta form registrasi
Data Uji		Data uji menggunakan data penulis
Dilanjutkan ke halaman selanjutnya		

Tabel 5.1 – lanjutan dari halaman sebelumnya

<b>ID</b>	<b>UJ-1-KP01</b>
Langkah pengujian	<ol style="list-style-type: none"> <li>1. Form registrasi diisi sesuai data uji</li> <li>2. Setelah selesai mengisi, mengklik tombol "Registrasi"</li> </ol>
Hasil yang Diharapkan	Sistem berhasil menyimpan data tersebut, <i>email</i> konfirmasi terkirim ke <i>mailbox email</i> yang didaftarkan
Hasil Pengujian	100% berhasil
<b>Skenario 2</b>	Menguji fitur <i>login</i>
Kondisi Awal	Sistem menampilkan form <i>login</i>
Data Uji	Data uji menggunakan data penulis yang sudah diregistrasi sebelumnya
Langkah pengujian	<ol style="list-style-type: none"> <li>1. Form tersebut diisi sesuai data uji</li> <li>2. Setelah selesai mengisi, mengklik tombol "Login"</li> </ol>
Hasil yang Diharapkan	Login berhasil
Hasil Pengujian	100% berhasil
<b>Skenario 3</b>	Menguji fitur konfirmasi <i>email</i>
Dilanjutkan ke halaman selanjutnya	

Tabel 5.1 – lanjutan dari halaman sebelumnya

ID	UJ-1-KP01
Kondisi Awal	Pengguna sedang membuka <i>email</i> dan mengklik <i>link</i> yang dikirimkan oleh sistem
Data Uji	Data uji menggunakan data penulis yang sudah diregistrasi sebelumnya
Langkah pengujian	Mengklik URL dari URL konfirmasi <i>email</i> yang dikirimkan oleh sistem
Hasil yang Diharapkan	<i>Email</i> berhasil dikonfirmasi
Hasil Pengujian	100% berhasil

### 5.1.2.2 Pengujian Fungsionalitas Manajemen Penawaran

Tabel 5.2 Pengujian Fungsionalitas Fitur Manajemen Penawaran

ID	UJ-1-KP02
<b>Referensi Kasus</b> <b>Penggunaan</b>	KP02
<b>Nama</b>	Pengujian fitur manajemen penawaran
<b>Skenario 1</b>	Menguji fitur melihat barang yang sedang aktif dilelang
Kondisi Awal	Sistem
Data Uji	Data uji menggunakan data penulis
Langkah pengujian	Membuka halaman website aplikasi lelang <i>online</i> via <i>browser</i> di alamat <a href="https://lelangapa.com">https://lelangapa.com</a>
Dilanjutkan ke halaman selanjutnya	

Tabel 5.2 – lanjutan dari halaman sebelumnya

<b>ID</b>	<b>UJ-1-KP02</b>
Hasil yang Diharapkan	Sistem berhasil menampilkan data barang yang sedang aktif dilelang
Hasil Pengujian	100% berhasil
<b>Skenario 2</b>	Menguji fitur mencari barang
Kondisi Awal	Sistem menampilkan halaman dengan elemen <i>input</i> search barang
Data Uji	
Langkah pengujian	<ol style="list-style-type: none"> <li>1. Elemen <i>input search</i> diisi dengan kata kunci “Jersey”</li> <li>2. Setelah selesai mengisi, mengklik tombol “Search”</li> </ol>
Hasil yang Diharapkan	Barang yang mengandung kata “Jersey” muncul dalam hasil pencarian
Hasil Pengujian	100% berhasil
<b>Skenario 3</b>	Menawar barang
Kondisi Awal	Pengguna sedang membuka halaman lelang barang
Data Uji	Data ujinya adalah memasukkan penawaran harga yang lebih tinggi pada penawaran harga saat itu
Langkah pengujian	
Dilanjutkan ke halaman selanjutnya	

Tabel 5.2 – lanjutan dari halaman sebelumnya

ID	UJ-1-KP02
	1. Memasukkan harga penawaran 2. Mengklik tombol “Tawar”
Hasil yang Diharapkan	Penawaran yang baru berhasil masuk ke dalam sistem, dan harga terbaru diupdate di halaman secara <i>realtime</i>
Hasil Pengujian	100% berhasil

### 5.1.2.3 Pengujian Fungsionalitas Manajemen Barang Lelang

Tabel 5.3 Pengujian Fungsionalitas Fitur Manajemen Barang Lelang

ID	UJ-1-KP03
<b>Referensi Kasus Penggunaan</b>	KP03
<b>Nama</b>	Pengujian fitur manajemen barang lelang
<b>Skenario 1</b>	Menguji fitur melihat barang yang sedang aktif dilelang
Kondisi Awal	Sistem
Data Uji	Data uji menggunakan data penulis
Dilanjutkan ke halaman selanjutnya	



Tabel 5.3 – lanjutan dari halaman sebelumnya

<b>ID</b>	<b>UJ-1-KP03</b>
Langkah pengujian	Membuka halaman website aplikasi lelang <i>online</i> via <i>browser</i> di alamat <a href="https://lelangapa.com">https://lelangapa.com</a>
Hasil yang Diharapkan	Sistem berhasil menampilkan data barang yang sedang aktif dilelang
Hasil Pengujian	100% berhasil
<b>Skenario 2</b>	Menguji fitur mencari barang
Kondisi Awal	Sistem menampilkan halaman dengan elemen <i>input</i> search barang
Data Uji	
Langkah pengujian	<ol style="list-style-type: none"> <li>1. Elemen <i>input search</i> diisi dengan kata kunci “Jersey”</li> <li>2. Setelah selesai mengisi, mengklik tombol “Search”</li> </ol>
Hasil yang Diharapkan	Barang yang mengandung kata “Jersey” muncul dalam hasil pencarian
Hasil Pengujian	100% berhasil
<b>Skenario 3</b>	Menawar barang
Kondisi Awal	Pengguna sedang membuka halaman lelang barang
Dilanjutkan ke halaman selanjutnya	

Tabel 5.3 – lanjutan dari halaman sebelumnya

<b>ID</b>	<b>UJ-1-KP03</b>
Data Uji	Data ujinya adalah memasukkan penawaran harga yang lebih tinggi pada penawaran harga saat itu
Langkah pengujian	<ol style="list-style-type: none"> <li>1. Memasukkan harga penawaran</li> <li>2. Mengklik tombol “Tawar”</li> </ol>
Hasil yang Diharapkan	Penawaran yang baru berhasil masuk ke dalam sistem, dan harga terbaru diupdate di halaman secara <i>realtime</i>
Hasil Pengujian	100% berhasil

#### 5.1.2.4 Pengujian Fungsionalitas Manajemen Interaksi Antarpengguna

Tabel 5.4 Pengujian Fungsionalitas Fitur Manajemen Interaksi Antarpengguna

<b>ID</b>	<b>UJ-1-KP04</b>
<b>Referensi Kasus Penggunaan</b>	KP04
<b>Nama</b>	Pengujian fitur manajemen interaksi antarpengguna
<b>Skenario 1</b>	Menguji fitur melihat review pengguna
Kondisi Awal	Sistem menampilkan halaman profil pengguna
Dilanjutkan ke halaman selanjutnya	

Tabel 5.4 – lanjutan dari halaman sebelumnya

<b>ID</b>	<b>UJ-1-KP04</b>
Data Uji	Data uji menggunakan data penulis
Langkah pengujian	Membuka halaman profil pengguna yang ingin dilihat <i>reviewnya</i> .
Hasil yang Diharapkan	Sistem berhasil menampilkan profil pengguna yang ingin dilihat <i>reviewnya</i> .
Hasil Pengujian	100% berhasil
<b>Skenario 2</b>	Menguji fitur mencari barang
Kondisi Awal	Sistem menampilkan halaman dengan elemen <i>input</i> search barang
Data Uji	
Langkah pengujian	<ol style="list-style-type: none"> <li>1. Elemen <i>input</i> search diisi dengan kata kunci “Jersey”</li> <li>2. Setelah selesai mengisi, mengklik tombol “Search”</li> </ol>
Hasil yang Diharapkan	Barang yang mengandung kata “Jersey” muncul dalam hasil pencarian
Hasil Pengujian	100% berhasil
<b>Skenario 3</b>	Menguji fitur berkirin pesan
Kondisi Awal	Pengguna sedang membuka halaman perpesanan kepada pengguna yang ingin dituju
Dilanjutkan ke halaman selanjutnya	

Tabel 5.4 – lanjutan dari halaman sebelumnya

ID	UJ-1-KP04
Data Uji	Data ujinya adalah pesan “Apakah barang X masih ada?”
Langkah pengujian	<ol style="list-style-type: none"> <li>1. Memasukkan data uji ke dalam elemen <i>input</i></li> <li>2. Mengklik tombol “Kirim Pesan”</li> </ol>
Hasil yang Diharapkan	Pesan berhasil terkirim dan <i>diupdate</i> di halaman yang sedang dibuka secara <i>realtime</i>
Hasil Pengujian	100% berhasil
<b>Skenario 4</b>	Menguji fitur melihat daftar pesan
Kondisi Awal	Sistem menampilkan halaman daftar pesan
Data Uji	-
Langkah pengujian	-
Hasil yang Diharapkan	Daftar pesan berhasil ditampilkan
Hasil Pengujian	100% berhasil

#### 5.1.2.5 Pengujian Fungsionalitas Manajemen Kupon

Tabel 5.5 Pengujian Fungsionalitas Fitur Manajemen Kupon

ID	UJ-1-KP05
<b>Referensi Kasus Penggunaan</b>	KP05
Dilanjutkan ke halaman selanjutnya	

Tabel 5.5 – lanjutan dari halaman sebelumnya

<b>ID</b>	<b>UJ-1-KP05</b>
<b>Nama</b>	Pengujian fitur manajemen kupon
<b>Skenario 1</b>	Menguji fitur melihat daftar kupon
Kondisi Awal	Sistem menampilkan halaman yang menampilkan halaman daftar kupon
Data Uji	-
Langkah pengujian	Membuka halaman profil pengguna yang ingin dilihat <i>reviewnya</i> .
Hasil yang Diharapkan	Sistem berhasil menampilkan profil pengguna yang ingin dilihat <i>reviewnya</i> .
Hasil Pengujian	100% berhasil
<b>Skenario 2</b>	Menguji fitur menambah kupon
Kondisi Awal	Sistem menampilkan halaman tambah kupon beserta formnya
Data Uji	Data-data kupon yang ingin ditambahkan
Langkah pengujian	<ol style="list-style-type: none"> <li>1. Mengisi <i>form</i> tambah kupon dengan informasi data uji</li> <li>2. Setelah selesai mengisi, mengklik tombol “Tambah”</li> </ol>
Dilanjutkan ke halaman selanjutnya	

Tabel 5.5 – lanjutan dari halaman sebelumnya

<b>ID</b>	<b>UJ-1-KP05</b>
Hasil yang Diharapkan	Kupon berhasil ditambahkan, dan muncul di halaman daftar tambah kupon
Hasil Pengujian	100% berhasil
<b>Skenario 3</b>	Menguji fitur memperbarui kupon
Kondisi Awal	Sistem menampilkan halaman perbarui kupon
Data Uji	Data ujinya adalah mengubah tanggal <i>expired</i> kupon menjadi 1 minggu setelahnya
Langkah pengujian	<ol style="list-style-type: none"> <li>1. Memasukkan data uji ke dalam <i>form</i></li> <li>2. Mengklik tombol “Perbarui”</li> </ol>
Hasil yang Diharapkan	Pesan berhasil terkirim dan diupdate di halaman yang sedang dibuka secara <i>realtime</i>
Hasil Pengujian	100% berhasil
<b>Skenario 4</b>	Menguji fitur menggunakan kupon
Kondisi Awal	Pengguna membuka halaman riwayat penawaran
Dilanjutkan ke halaman selanjutnya	

Tabel 5.5 – lanjutan dari halaman sebelumnya

ID	UJ-1-KP05
Data Uji	Kupon “SKRIPSIBAHAGIA” yang mendiskon harga sebesar 20% dan “AKULULUS” yang merupakan kupon <i>freeshipping</i>
Langkah pengujian	<ol style="list-style-type: none"> <li>1. Memasukkan data uji ke dalam elemen <i>input</i> nama kupon</li> <li>2. Mengklik tombol “Gunakan kupon”</li> </ol>
Hasil yang Diharapkan	Kupon berhasil digunakan, dan muncul <i>modal</i> yang menandakan kesuksesan penggunaan kupon
Hasil Pengujian	100% berhasil

### 5.1.3 Pengujian Kecepatan

#### 5.1.3.1 Lingkungan Pengujian

Lingkungan pengujian kecepatan sistem pada pengerjaan Tugas Akhir ini dilakukan pada perangkat keras sebagai berikut:

- a. Jenis Perangkat: Laptop
- b. Sistem Operasi: Windows
- c. *Browser*: Chrome
- d. Ukuran Layar: 1920 x 1280
- e. RAM: 6GB
- f. Kecepatan akses internet: 150Kbps

### 5.1.3.2 Hasil Pengujian

**Tabel 5.6** Pengujian Fungsionalitas Fitur Manajemen Interaksi Antarpengguna

ID Usecase	Ukuran Halaman	DOM Loading (dalam ms)	Scripting (dalam ms)	Render (dalam ms)	Total
UC01-01	625KB	77,1	600,9	119,2	797,2
UC01-02	800KB	101	540	80	721
UC02-01	12,4MB	146,2	5592,8	1860,9	7599,0
UC02-02	12,4MB	150,7	3000	3512	8192,7
UC02-03	1,5MB	122,9	1119,3	438	1680,2
UC02-04	1,2MB	91,6	909,7	327,9	1329,2
UC03-01	1,5MB	109	1277	369	1755
UC03-02	1,6MB	90	1329	420	1839
UC04-01	1,6MB	165,5	826,7	559,6	1551,8
UC04-02	1,2MB	90,7	917,8	219,8	1228,3
UC04-03	1,5MB	122,9	1119,3	438	1680,2
UC04-04	1,2MB	95,5	629,9	269,7	995,1
Dilanjutkan ke halaman selanjutnya					



Tabel 5.6 – lanjutan dari halaman sebelumnya

ID Usecase	Ukuran Halaman	DOM Loading (dalam ms)	Scripting (dalam ms)	Render (dalam ms)	Total
UC04-05	1,6MB	118,1	915,9	380,0	1410,9
UC05-04	1,6MB	80,9	791,1	210,3	1082,3
<b>Rata-rata</b>		104	914	313	1332,6

### 5.1.4 *User Experience Assesment*

#### 5.1.4.1 Deskripsi Pengujian

Pengujian dilakukan dengan menggunakan kriteria yang mengacu kepada penelitian *Development of an Instrument Measuring User Satisfaction of the Human-Computer Interface*[31], yaitu sebagai berikut:

1. Desain & Impresi *Web*
2. Kejelasan & Konsistensi Sistem
3. Kemudahan Penggunaan
4. Kejelasan status proses
5. *Error message* yang jelas
6. Kecepatan
7. *Rating* Keseluruhan
8. Tingkat rekomendasi pada teman

Pengujian ini dilakukan pada tanggal 8 Juli 2017 di Kebun Bibit Wonorejo, dengan secara sembarang memilih responden yang sedang ada di kebun tersebut. Kemudian, responden diminta untuk menggunakan aplikasi yang sudah ada sebelumnya (balelang.com) dan aplikasi tugas akhir ini selama beberapa menit, dan setelah dirasa cukup, lalu responden mengisi kuisioner yang telah diberikan.

#### 5.1.4.2 Hasil Pengujian

Hasil rekapitulasi kuisioner *user experience* dapat dilihat pada Tabel 5.7.

**Tabel 5.7** Rekapitulasi Hasil Pengujian Pengguna Aplikasi Lelang *Online*

<b>Parameter</b>	<b>Rata-rata Nilai Aplikasi Lain</b>	<b>Rata-rata Nilai Aplikasi Lelangapa</b>	<b>Persentase Perbedaan</b>
Desain & Impresi Web	3,3	4,1	20% (meningkat)
Kejelasan & konsistensi sistem	3,5	4,2	17% (meningkat)
Kemudahan penggunaan	3,1	3,9	21% (meningkat)
Kejelasan status proses	3,7	3,9	5% (meningkat)
Error message yang jelas	3,3	4	18% (meningkat)
Performa	3,7	3,8	3% (meningkat)
Rating keseluruhan	3,7	4,3	14% (meningkat)
Rekomendasi aplikasi pada teman?	3,4	4,3	21% (meningkat)

### 5.1.5 *Maintainability Assesment*

#### 5.1.5.1 Deskripsi Pengujian

Pengujian dilakukan dengan menggunakan kriteria yang mengacu kepada paper *A Software Maintainability Evaluation Methodology*[32]. Paper ini menyebutkan bahwa ada 2 kriteria dasar penilaian *maintainability* dengan *weight* berbeda, yaitu dokumentasi (*weight*: 40%) dan *source code*(*weight*: 60%). Menurut paper tersebut, terdapat 5 faktor utama *maintainability*, yaitu sebagai berikut:

1. *modularity*;
2. *descriptiveness*;
3. *consistency*;
4. *simplicity*; dan
5. *trackability*

Pengujian ini dilakukan dengan cara menyebar *form online* dengan menggunakan media Google Form. Responden yang disasar adalah responden dengan latar belakang *software engineer* dengan tujuan agar responden dapat membandingkan pengalaman responden tersebut dengan kualitas *maintainability* tugas akhir ini.

#### 5.1.5.2 Hasil Pengujian

Hasil rekapitulasi kuisioner *maintainability* untuk aspek *source code* dapat dilihat pada Tabel 5.8.

**Tabel 5.8** Rekapitulasi Pengujian *Maintainability* pada Aspek Kode Sumber

Kriteria	R1	R2	R3	R4	R5
<i>Modularity</i>	4	5	3	5	4
<i>Descriptiveness</i>	4	4	2	5	5
<i>Consistency</i>	4	4	3	5	5
dilanjutkan ke halaman setelahnya					

Tabel 5.8 – dilanjutkan dari halaman sebelumnya

<b>Kriteria</b>	R1	R2	R3	R4	R5
<i>Simplicity</i>	4	5	3	5	3
<i>Trackability</i>	4	4	2	5	3

Hasil rekapitulasi kuisioner *maintainability* untuk aspek *source code* dapat dilihat pada Tabel 5.9.

**Tabel 5.9** Rekapitulasi Pengujian *Maintainability* pada Aspek Dokumentasi

<b>Kriteria</b>	R1	R2	R3	R4	R5
<i>Modularity</i>	3	3	3	3	3
<i>Descriptiveness</i>	4	4	2	4	4
<i>Consistency</i>	4	4	3	4	4
<i>Simplicity</i>	5	5	3	5	5
<i>Trackability</i>	4	5	2	5	3

## 5.2 Evaluasi

Pada subbab ini, penulis akan memaparkan hasil analisa terhadap aplikasi, perspektif non-IT terhadap pengerjaan maupun lingkup pekerjaan dari aplikasi Lelang *online* ini.

### 5.2.1 Pendekatan Hukum Perlindungan Konsumen

Sebagaimana lelang *online* adalah salah satu jenis dari jenis transaksi jual-beli barang, tentu saja dalam pelaksanaannya diatur oleh undang-undang dan diawasi pemerintah, terutama untuk lelang harta-harta berharga seperti surat tanah, mobil, ijin usaha, dan lain-lain. Namun, penulis sadar banyak sekali kekurangan pengkajian hukum dan peraturan-peraturan penting seperti hak-hak dan kewajiban masing-masing pihak, *rule* agar lelang dapat berjalan dan diawasi dengan baik, dan peraturan

lainnya

Untuk mempermudah pembahasan dan agar pemaparannya lebih kredibel (karena penulis tidak *capable* dan kredibel untuk memaparkan hal ini), maka penulis mengutip sebuah kasus penipuan *online* - dalam sebuah pertanyaan di platform konsultasi hukum online. *Platform online* ini - hukumonline.com. Pada platform ini, sering terdapat kajian kasus-kasus hukum dan menggunakan pendekatan hukum untuk penyelesaiannya. Dalam kasus ini, setelah berdiskusi dengan teman dan saudara yang mengambil spesialisasi hukum, penulis dengan hati-hati memaparkan sesuai dengan poin-poin yang persis ada dalam forum *online* tersebut, dan tidak mengubah satupun kata agar kebenaran informasi yang disampaikan tidak berubah.

Pada keterangannya, pengkajian kasus ini menggunakan pendekatan utama pada **Undang-Undang Nomor 8 Tahun 1999 tentang Perlindungan Konsumen ("UU PK")** dan **Peraturan Pemerintah Nomor 82 Tahun 2012 tentang Penyelenggaraan Sistem dan Transaksi Elektronik ("PP PSTE")**. PP PSTE sendiri merupakan turunan dari **Undang-Undang Nomor 11 Tahun 2008 tentang Informasi dan Transaksi Elektronik ("UU ITE")**.

#### **5.2.1.1 Perlindungan Hukum Bagi Konsumen Belanja Online**

Pemaparan subbab ini berupa pertanyaan yang dikaji dalam hukumonline.com, disajikan dalam bentuk pertanyaan dan jawaban.

##### **Pertanyaan**

Saya pernah belanja barang secara online, tapi barang yang saya beli tidak sama dengan yang saya lihat di foto pada iklan yang dipajang. Pertanyaan

saya, apakah itu termasuk pelanggaran hak konsumen? Apakah saya dapat menuntut penjual untuk mengembalikan uang atau mengganti barang yang saya beli tersebut? Terima kasih.

**Jawaban oleh hukumonline.com:**

Dengan pendekatan UU Perlindungan Konsumen, kasus yang Anda sampaikan tersebut dapat kami simpulkan sebagai salah satu pelanggaran terhadap hak konsumen.

Pasal 4 UU PK menyebutkan bahwa hak konsumen/dalam kasus ini adalah pengguna aplikasi lelang *online* adalah sebagai berikut:

- a. hak atas kenyamanan, keamanan, dan keselamatan dalam mengkonsumsi barang dan/atau jasa;
- b. hak untuk memilih barang dan/atau jasa serta mendapatkan barang dan/atau jasa tersebut sesuai dengan nilai tukar dan kondisi serta jaminan yang dijanjikan;
- c. **hak atas informasi yang benar, jelas, dan jujur mengenai kondisi dan jaminan barang dan/atau jasa;**
- d. hak untuk didengar pendapat dan keluhannya atas barang dan/atau jasa yang digunakan;
- e. **hak untuk mendapatkan advokasi, perlindungan, dan upaya penyelesaian sengketa perlindungan konsumen secara patut;**
- f. hak untuk mendapat pembinaan dan pendidikan konsumen;
- g. hak untuk diperlakukan atau dilayani secara benar dan jujur serta tidak diskriminatif;
- h. **hak untuk mendapatkan kompensasi, ganti rugi dan/atau penggantian, apabila barang dan/atau jasa yang diterima tidak sesuai dengan perjanjian atau tidak sebagaimana mestinya;**
- i. hak-hak yang diatur dalam ketentuan peraturan perundangundangan lainnya.

Di sisi lain, kewajiban bagi pelaku usaha (dalam hal ini adalah

penjual online), sesuai Pasal 7 UU PK adalah:

- a. beritikad baik dalam melakukan kegiatan usahanya;
- b. **memberikan informasi yang benar, jelas dan jujur mengenai kondisi dan jaminan barang dan/atau jasa serta memberi penjelasan penggunaan, perbaikan dan pemeliharaan;**
- c. memperlakukan atau melayani konsumen secara benar dan jujur serta tidak diskriminatif;
- d. **menjamin mutu barang dan/atau jasa yang diproduksi dan/atau diperdagangkan berdasarkan ketentuan standar mutu barang dan/atau jasa yang berlaku;**
- e. memberi kesempatan kepada konsumen untuk menguji, dan/atau mencoba barang dan/atau jasa tertentu serta memberi jaminan dan/atau garansi atas barang yang dibuat dan/atau yang diperdagangkan;
- f. **memberi kompensasi, ganti rugi dan/atau penggantian atas kerugian akibat penggunaan, pemakaian dan pemanfaatan barang dan/atau jasa yang diperdagangkan;**
- g. **memberi kompensasi, ganti rugi dan/atau penggantian apabila barang dan/atau jasa yang diterima atau dimanfaatkan tidak sesuai dengan perjanjian.**

Selaku konsumen sesuai Pasal 4 huruf h UU PK tersebut, berhak mendapatkan kompensasi, ganti rugi dan/atau penggantian apabila barang dan/atau jasa yang diterima tidak sesuai dengan perjanjian atau tidak sebagaimana mestinya. Sedangkan, pelaku usaha itu sendiri sesuai Pasal 7 huruf (g) UU PK berkewajiban memberi kompensasi, ganti rugi dan/atau penggantian apabila barang dan/atau jasa yang diterima atau dimanfaatkan tidak sesuai dengan perjanjian.

Apabila pelaku usaha (dalam hal ini, penulis karena penulis sebagai perantara antara peledang dan pembeli) tidak

melaksanakan kewajibannya, pelaku usaha dapat dipidana berdasarkan Pasal 62 UUPK, yang berbunyi:

”Pelaku usaha yang melanggar ketentuan sebagaimana dimaksud dalam Pasal 8, Pasal 9, Pasal 10, Pasal 13 ayat (2), Pasal 15, Pasal 17 ayat (1) huruf a, huruf b, huruf c, huruf e, ayat (2) dan Pasal 18 dipidana dengan **pidana penjara paling lama 5 (lima) tahun** atau pidana denda paling banyak **Rp 2.000.000.000,00 (dua milyar rupiah)**”.

#### **5.2.1.2 Kontrak Elektronik dan Perlindungan Konsumen berdasarkan UU ITE dan PP PSTE**

Transaksi jual beli Anda, meskipun dilakukan secara online, berdasarkan UU ITE dan PP PSTE tetap diakui sebagai transaksi elektronik yang dapat dipertanggungjawabkan. Persetujuan konsumen untuk membeli barang secara *online* dengan cara melakukan klik persetujuan atas transaksi merupakan bentuk tindakan penerimaan yang menyatakan persetujuan dalam kesepakatan pada transaksi elektronik. Tindakan penerimaan tersebut biasanya didahului pernyataan persetujuan atas syarat dan ketentuan jual beli secara *online* yang dapat kami katakan juga sebagai salah satu bentuk **Kontrak Elektronik**. Kontrak Elektronik menurut **Pasal 47 ayat (2) PP PSTE** dianggap sah apabila:

- a. terdapat kesepakatan para pihak; dilakukan oleh subjek hukum yang cakap atau yang berwenang mewakili sesuai dengan ketentuan peraturan
- b. perundang-undangan;
- c. terdapat hal tertentu; dan
- d. objek transaksi tidak boleh bertentangan dengan peraturan perundang-undangan, kesusilaan, dan ketertiban umum.

Kontrak Elektronik itu sendiri menurut **Pasal 48 ayat (3) PP PSTE** setidaknya harus memuat hal-hal sebagai berikut:



- a. data identitas para pihak;
- b. objek dan spesifikasi;
- c. persyaratan Transaksi Elektronik;
- d. harga dan biaya;
- e. prosedur dalam hal terdapat pembatalan oleh para pihak;
- f. ketentuan yang memberikan hak kepada pihak yang dirugikan untuk dapat mengembalikan barang dan/atau meminta penggantian produk jika terdapat cacat tersembunyi; dan
- g. pilihan hukum penyelesaian Transaksi Elektronik.

Terkait dengan perlindungan konsumen, **Pasal 49 ayat (1) PP PSTE menegaskan bahwa Pelaku Usaha yang menawarkan produk melalui Sistem Elektronik wajib menyediakan informasi yang lengkap dan benar berkaitan dengan syarat kontrak, produsen, dan produk yang ditawarkan.**

### 5.2.1.3 Kesimpulan

Untuk mencapai hasil pengujian yang maksimal, seharusnya aplikasi lelang *online* ini melakukan tes pasar/tes pengguna, dimana aplikasi ini benar-benar *dilempar* ke pasar, lewat iklan/*endorsements/digital advertising* lainnya. Penulis sudah mempersiapkan aplikasi untuk dapat melakukan tes pasar, yaitu *hosting* aplikasi, pembelian domain sedemikian rupa agar aplikasi ini dapat diakses masyarakat luas dan benar-benar siap untuk tes pasar.

Ppenulis sadar bahwa dalam proses pengerjaan maupun saat *requirement gathering*, sangat sedikit pertimbangan hukum (seperti peraturan lelang yang adil, proses pengawasan lelang yang sesuai dengan peraturan pemerintah, dll) dimasukkan dalam proses pembuatan aplikasi, sementara proses jual-beli itu sendiri proses yang vital dan hak kewajiban masing-masing pelaku (baik pembeli, penjual dan ) diatur dalam undang-undang.

Poin-poin yang dicetak tebal pada pemaparan pasal 4 dan

pasal 7 UU PK adalah poin-poin yang tidak bisa penulis penuhi dalam posisi pelaku usaha dalam pengerjaan tugas akhir ini. Terkait dengan sanksi dan pidana yang menunggu jika penulis tetap menjalankan tes pasar tanpa mengikuti undang-undang yang ada, inkapabilitas penulis dalam sebagai penyelenggara usaha (lelang *online*), maka penulis memutuskan untuk tidak melakukan tes pasar.

Dasar hukum yang digunakan (tetap dikutip dari hukumonline.org):

1. Kitab Undang-Undang Hukum Pidana
2. Undang-Undang Nomor 8 Tahun 1999 tentang Perlindungan Konsumen
3. Undang-Undang Nomor 11 Tahun 2008 tentang Informasi dan Transaksi Elektronik
4. Peraturan Pemerintah Nomor 82 Tahun 2012 tentang Penyelenggaraan Sistem dan Transaksi Elektronik

### 5.2.2 Evaluasi Pengujian *User Experience*

Rekapitulasi tersebut diurutkan berdasarkan besarnya/signifikansi persentase perbedaan antara kedua platform, dapat dilihat pada Tabel 5.10.

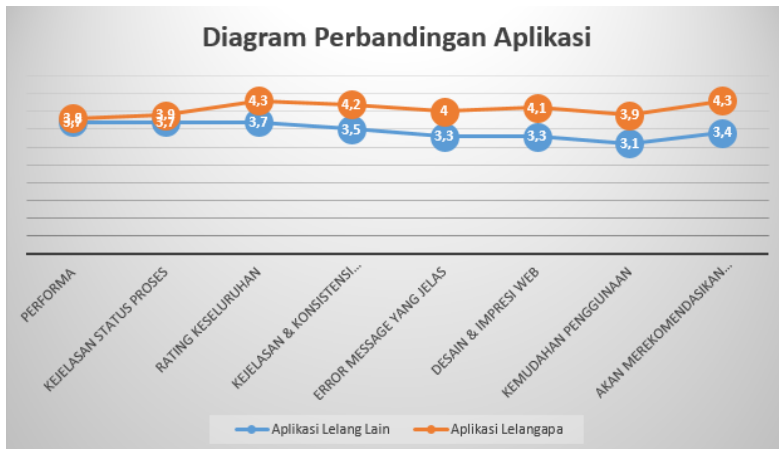
**Tabel 5.10** Rekapitulasi Perbedaan Hasil Pengujian Pengguna Aplikasi Lelang *Online*

Parameter	Persentase Perbedaan
Performa	3%
Kejelasan status proses	5%
<i>Rating</i> keseluruhan	14%
Kejelasan & konsistensi sistem	17%
dilanjutkan ke halaman selanjutnya	

Tabel 5.10 – dilanjutkan dari halaman sebelumnya

Parameter	Persentase Perbedaan
Error message yang jelas	18%
Desain & Impresi Web	20%
Kemudahan penggunaan	21%
Akan merekomendasikan aplikasi ini pada teman?	21%

Visualisasi perbandingan dapat dilihat lewat diagram garis pada Gambar 5.1.



**Gambar 5.1** Diagram Perbandingan Pengujian *User Experience* Pengguna

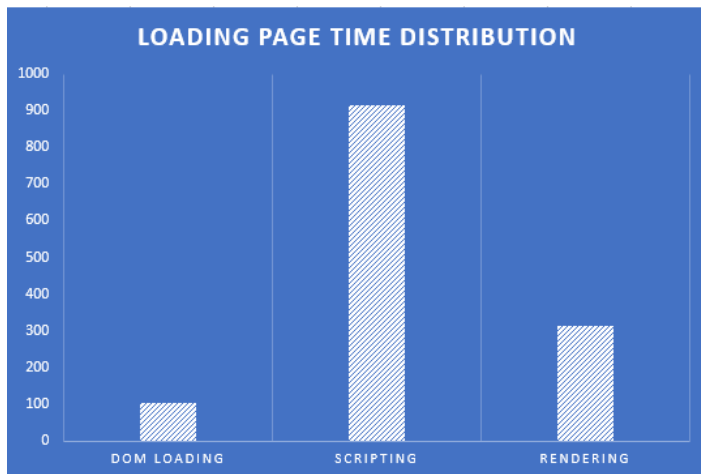
Dalam hal ini, dapat disimpulkan bahwa impresi *user experience* sudah baik, dan skor *user experience*nya sedikit diatas sistem serupa lainnya. Perbedaannya sudah cukup signifikan adalah *recommendation*, kemudahan penggunaan dan desain *web* yang baik. Namun, yang menjadi perhatian adalah perbedaan *performa* yang masih sangat kecil. Hal ini terkait dengan pengujian *speed test* di subbab selanjutnya.

### 5.2.3 Evaluasi Pengujian Kecepatan

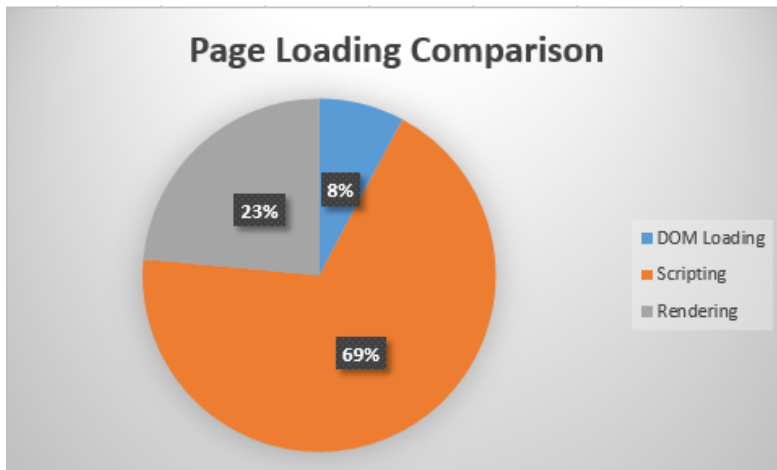
Hasil rata-rata dari kecepatan *loading* dan *request* terhadap sistem adalah sebagai berikut, sesuai dengan segmentasi yang telah dipaparkan pada subbab Pengujian Kecepatan:

1. *DOM Loading*: 104,2 ms
2. *Scripting*: 914,6 ms
3. *Rendering*: 313,7 ms

Visualisasi perbandingan/komparasi antara ketiga segmentasi tersebut dapat dilihat pada Gambar 5.1.



**Gambar 5.2** Diagram Batang Hasil Pengujian Kecepatan Sistem



**Gambar 5.3** Diagram Lingkaran Hasil Pengujian Kecepatan

Dari angka tersebut, dapat dilihat bahwa proses *scripting* sangat memakan waktu, yang berarti konten dan *assets* yang dimuat dalam page cukup besar. Tampak dari waktu *loading page* halaman menampilkan daftar barang mencapai waktu *loading* terlama. Dengan menggunakan *tool* evaluasi Lighthouse, tampak bahwa *loading* gambar sangat memakan waktu. Selain itu, adanya beberapa *script* dibutuhkan yang kurang efisien (butuh *owl-carousel*, *bootstrap* untuk memuat setiap halaman).

#### 5.2.4 Evaluasi Pengujian *Maintainability*

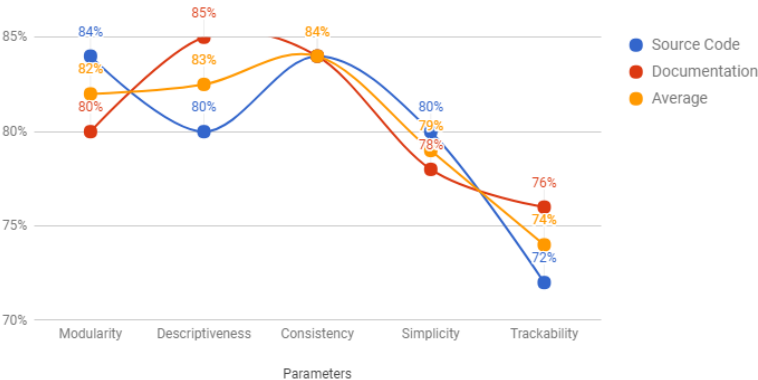
Sesuai dengan daftar pengujian yang telah dilakukan pada subbab Pengujian *Maintainability*, maka dapat dievaluasi sesuai dengan Tabel 5.11. Visualisasi perbandingan dapat dilihat lewat diagram garis pada Gambar 5.1. Dengan total skor 80% yang tepat sama dengan *evaluator goal* yang disebutkan pada subbab

Deskripsi Pengujian *Maintainability*, maka dapat dikatakan bahwa *maintainability* pada sistem ini sudah cukup baik.

Tabel 5.11 Tabel Evaluasi Hasil Pengujian *Maintainability*

Parameter	Kode Sumber	Dokumentasi	Rata-rata
<i>Modularity</i>	84%	80%	82%
<i>Descriptiveness</i>	80%	85%	83%
<i>Consistency</i>	84%	84%	84%
<i>Simplicity</i>	80%	78%	79%
<i>Trackability</i>	72%	76%	74%
<b>Rata-rata</b>	80%	81%	80%
<i>Weight</i>	60%	40%	
<i>Score</i>	48%	32%	
<i>Final Score</i>	<b>80%</b>		

Maintainability Evaluation Diagram



Gambar 5.4 Diagram Evaluasi Pengujian *Maintainability*

### 5.2.5 *Summary* Evaluasi

*Summary* dari evaluasi pengujian dapat dilihat pada Tabel 5.12.

**Tabel 5.12** Evaluasi Keseluruhan

<b>Jenis Pengujian</b>	<b>Skor Target</b>	<b>Skor Tercapai</b>	<b>Kesimpulan</b>
Fungsionalitas	Semua teruji	Semua teruji	Berhasil
Kecepatan	3s	3,2s	Tidak tercapai, namun selisih keduanya tidak terlalu jauh
<i>Maintainability</i>	80%	80%	Berhasil
<i>User Experience</i>	-	+15% on average	Baik

*(Halaman ini sengaja dikosongkan)*



## BAB VI

### PENUTUP

Bab ini membahas kesimpulan yang dapat diambil dari tujuan pembuatan sistem dan hubungannya dengan hasil uji coba dan evaluasi yang telah dilakukan. Selain itu, terdapat beberapa saran yang bisa dijadikan acuan untuk melakukan pengembangan dan penelitian lebih lanjut.

#### 6.1 Kesimpulan

Dari proses perancangan, implementasi dan pengujian terhadap sistem, dapat diambil beberapa kesimpulan berikut:

1. Kualitas perancangan dan desain sistem dan fleksibilitas sistem sangat penting dalam rancang bangun aplikasi jual-beli online, karena sifat perubahan yang sangat cepat.
2. *User Experience* adalah faktor yang sangat penting dalam kesuksesan *platform* jual-beli online.
3. Selain *user experience*, *maintainability* juga sangat penting dalam pengembangan perangkat lunak.

#### 6.2 Saran

Berikut beberapa saran yang diberikan untuk pengembangan lebih lanjut:

1. Mengikutsertakan pihak yang *capable*/kredibel dan ahli di bidang hukum dan *bussiness process* untuk menetapkan alur, memperbaiki alur dan membuat alur monitoring untuk proses lelang yang lebih aman, kredibel.
2. Mempelajari platform lelang *online* di luar negeri yang sudah sukses, yakni mempelajari ide-ide, alur aktivitas dan penggunaan kaidah *user experience* dan *usability* dalam website tersebut dan dampaknya terhadap *revenue*.

*(Halaman ini sengaja dikosongkan)*

## DAFTAR PUSTAKA

- [1] Y.-F. Kuo, S.-T. Yen, and L.-H. Chen, “Online auction service failures in Taiwan: Typologies and recovery strategies,” *Science Direct*, vol. 10, pp. 183–193, Oct. 2016.
- [2] “Sistem Lelang Online Masih Bermasalah.” [Online]. Available: <http://www.saidabdullah.info/index.php/issues/394-sistem-lelang-online-masih-bermasalah>
- [3] Wikipedia, “Amazon Web Services,” Jan. 2016. [Online]. Available: [https://id.wikipedia.org/wiki/Amazon\\_Web\\_Services](https://id.wikipedia.org/wiki/Amazon_Web_Services)
- [4] A. Web Services, “Amazon Benefits.” [Online]. Available: <https://aws.amazon.com/application-hosting/benefits/>
- [5] “Sistem Basis Data,” Mar. 2013. [Online]. Available: <https://rezapratama8.wordpress.com/2013/05/19/sistem-basis-data/>
- [6] E. T. insideBIGDATA, “Exponential Growth of Data,” Feb. 2017. [Online]. Available: <https://insidebigdata.com/2017/02/16/the-exponential-growth-of-data/>
- [7] D. D. VanHoose, *E-Commerce Economics*, 2nd ed. Routledge, Mar. 2011. [Online]. Available: <https://books.google.co.id/books?id=doKsAgAAQBAJ&hl=id>
- [8] “JWT,” Jun. 2016. [Online]. Available: <http://blog.nostratech.com/2016/06/jwt-json-web-token.html>
- [9] “Apa itu Laravel?” 2016. [Online]. Available: <http://emka.web.id/tutorial/tutorial-laravel/2013/tutorial-laravel-1-apa-itu-laravel/>
- [10] Laravel, “Browser Testing.” [Online]. Available: <https://laravel.com/docs/5.4/dusk>

- [11] BalaiLelang, “Sejarah Lelang di Indonesia.” [Online]. Available: <https://www.balailelang.co.id/index.php/sejarah-lelang/sejarah-lelang-di-indonesia>
- [12] R. Pratama, “Lelang,” Feb. 2012. [Online]. Available: <http://raypratama.blogspot.co.id/2012/02/lelang.html>
- [13] Wikipedia, “Lelang,” Jan. 2016. [Online]. Available: <https://id.wikipedia.org/wiki/Lelang>
- [14] “Arti pengertian BID, INC, BIN dalam Bahasa Lelang Website.” [Online]. Available: <http://www.bocahsoenyi.com/2015/02/arti-pengertian-bid-inc-bin-dalam-lelang-website.html>
- [15] “MongoDB,” Jan. 2017. [Online]. Available: <https://id.wikipedia.org/wiki/MongoDB>
- [16] “Node.js (Wikipedia),” Nov. 2014. [Online]. Available: <https://id.wikipedia.org/wiki/Node.js>
- [17] Wikipedia, “NoSQL.” [Online]. Available: <https://id.wikipedia.org/wiki/NoSQL>
- [18] A. Azurri, “Node Package Manager,” Oct. 2016. [Online]. Available: <http://www.azuwir.com/2016/10/nodejs-node-package-manager-npm/>
- [19] “PostgreSQL.” [Online]. Available: <http://www.belajarsql.com/postgresql>
- [20] “Repository Pattern - Coding4ever’s Blog,” Feb. 2016. [Online]. Available: <http://coding4ever.net/blog/2016/02/22/repository-pattern/>
- [21] M. Shaw, “Some Patterns for Software Architectures,” *Pattern Languages of Program*

- Design*, vol. 2, pp. 255–269, 1996. [Online]. Available: [https://sophia.javeriana.edu.co/~cbustaca/docencia/DEAS-2015-03/documentos/1996\\_Shaw\\_Some%20Patterns%20for%20Software%20Architectures.pdf](https://sophia.javeriana.edu.co/~cbustaca/docencia/DEAS-2015-03/documentos/1996_Shaw_Some%20Patterns%20for%20Software%20Architectures.pdf)
- [22] Wikipedia, “SendGrid,” Mar. 2017. [Online]. Available: <https://en.wikipedia.org/wiki/SendGrid>
- [23] W. F. Google Developers, “Service Worker : Pengantar,” Apr. 2017. [Online]. Available: <https://developers.google.com/web/fundamentals/getting-started/primers/service-workers?hl=id>
- [24] “SMTP.” [Online]. Available: <http://techscape.co.id/hosting/smtp.ts>
- [25] “Socket.io (Wikipedia),” Jan. 2016. [Online]. Available: <https://en.wikipedia.org/wiki/Socket.IO>
- [26] W. , “Test Script,” Jan. 2016, translated by author. [Online]. Available: [https://en.wikipedia.org/wiki/Test\\_script](https://en.wikipedia.org/wiki/Test_script)
- [27] Wikipedia, “Vue.js,” May 2017, translated to Indonesia. [Online]. Available: <https://en.wikipedia.org/wiki/Vue.js>
- [28] —, “Lelang,” Mar. 2017. [Online]. Available: <https://id.wikipedia.org/wiki/Lelang>
- [29] T. G. Laia Bonastre, “A set of heuristics for user experience evaluation in e-commerce websites,” *ACHI 2014 : The Seventh International Conference on Advances in Computer-Human Interactions*, pp. 27–34, 2014.
- [30] I. Sommerville, *Software Engineering*, 9th ed. United States of America: Addison-Wesley. All rights reserved, Oct. 2009.

- [31] J. P. Chin, V. A. Diehl, and K. L. Norman, "Development of an Instrument Measuring User Satisfaction of the Human-Computer Interface," *Proceedings of ACM CHI'88 Conference on Human Factors in Computing Systems*, May 1998.
- [32] D. E. PEERCY, "A Software Maintainability Evaluation Methodology," *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING*, vol. SE-7, no. 4, pp. 343–351.

## BAB A

### KODE SUMBER

```
1
2 var fs = require('fs');
3 var app = require('express')();
4 var options = {
5     key: fs.readFileSync('/etc/nginx/ssl/
        ↪ lelangapa-socket.key'),
6     cert: fs.readFileSync('/etc/nginx/ssl/
        ↪ lelangapa-socket.crt')
7 };
8 var httpserver = require('https').Server(options
    ↪ , app);
9 var httpserver = require('http').Server(app);
10 //var io = require('socket.io')(server);
11 var ioServer = require('socket.io');
12 var biddingAPI = require('./bid_api/Bidding');
13 var peringkatAPI = require('./bid_api/Ranking');
14 var auctioneerAPI = require('./bid_api/Auctioneer
    ↪ ');
15 var pushnotifAPI = require('./notification_api/
    ↪ PushNotification');
16
17 var io = new ioServer();
18 var jwt = require('socketio-jwt');
19
20 /*
21  * environment integrated file for security
    ↪ issues
22  */
23 require('dotenv').config({path: '/home/grkt/
    ↪ security/.env'});
24
25 app.disable('etag');
26 app.get('/', function(req, res){
27     res.status(200).json({
28         status: "success"
29     });
```

```

30 });
31
32 httpsserver.listen(15000, function() {
33     console.log ("Connected on port 15000");
34 });
35 httpserver.listen(16000, function() {
36     console.log ("Connected on port 16000");
37 });
38 io.attach(httpsserver);
39 io.attach(httpserver);
40
41 io.use(jwt.authorize({
42     secret:process.env.JWT_SECRET,
43     handshake : true
44 }));
45
46 io.on('connection', function(socket){
47     /*
48     mengirim data bahwa berhasil konek
49     */
50     socket.emit('connected', { status: "true" });
51
52     /*
53     socket ketika ada yang menekan tombol "Tawar"
54     */
55     socket.on('submitting', function(room) {
56         //kirirkan ke semua client pada sebuah room,
57         ↪ kecuali sender
58         socket.to(room).emit('submitting', { status:
59         ↪ "true" });
60     });
61
62     /*
63     socket ketika ada yang berhenti mengetik
64     ↪ penawaran
65     */
66     socket.on('type-stop', function(room) {

```



```

65 //implemented later
66 io.to(room).emit('type-stop', { status: "true
    ↳ " });
67 });
68
69 /*
70 socket untuk memasukkan tawaran
71 */
72 socket.on('submitbid', function(bidjson) {
73     /* terima dari client berupa json yang berisi
74      * id_bidder, id_item, dan harga_bid
75      * lalu diteruskan ke API submitBid
76      */
77     var bidobj = JSON.parse(bidjson);
78     biddingAPI.submitBid(bidobj, function(status,
    ↳ result) {
79         if (status=="1")
80         {
81             //jika sukses, maka bid dari user akan
    ↳ dibroadcast ke semua user yang
    ↳ tersambung ke socket
82     peringkatAPI.getPeringkatTopThreeLive(
    ↳ bidobj.id_item, function(
    ↳ status_peringkat, result_peringkat)
    ↳ {
83         if (status_peringkat == "1")
84         {
85             var messageObject = {};
86             var tokenArray;
87             /*constructing message object)
88             result.peringkat = result_peringkat;
89             io.to(result.item_id_return)
90             .emit('bidsuccess', result);
91             pushnotifAPI.sendPushNotification(
    ↳ messageObject, tokenArray);
92         }
93     });
94 }

```

```

95         else if (status=="0")
96         {
97             //jika gagal, maka send ke sender bahwa
98             ↪ bid failed
99             socket.emit('bidfailed', { bidstatus: "
100             ↪ failed" });
101         }
102     });
103     //io.emit('biddingresponse', bidobj.msg);
104     //implemented later
105 }
106 /*
107 socket untuk membatalkan lelang
108 yang sedang atau yang akan berjalan
109 */
110 socket.on('cancelauction', function(
111     ↪ item_id_query) {
112     var room = item_id_query;
113     auctioneerAPI.cancelItemOnAuction(
114     ↪ item_id_query, function(status, result)
115     ↪ {
116         if (status=="1") {
117             io.to(room).emit('cancelsuccess', result)
118             ↪ ;
119         }
120         else if (status=="0") {
121             socket.emit('cancelfailed', result);
122         }
123     });
124 });
125 /*
126 socket untuk memilih pemenang lelang -- tested
127 */
128 socket.on('winnersselected', function(winnerjson
129     ↪ ) {
130     //JSON berisi item_id_query dan bid_id_query
131     var winnerObj = JSON.parse(winnerjson);

```

```

126     var room = winnerObj.item_id_query;
127     auctioneerAPI.stopAndSelectBidderWinner(
        ↪ winnerObj.bid_id_query, function(status
        ↪ , result) {
128         if (status=="1") {
129             io.to(room).emit('winnerchosen', result);
130         }
131         else if (status=="0") {
132             socket.emit('selectionfailed', result);
133         }
134     });
135 });
136
137     /*
138     socket untuk block pengguna pada suatu
        ↪ lelang --tested
139     */
140     socket.on('blockuser', function(
        ↪ jsonobject) {
141         var obj = JSON.parse(jsonobject);
142         var room = obj.id_item;
143         auctioneerAPI.blockUserOnBidding(obj,
        ↪ function(status, result) {
144             if (status == "1") {
145                 io.to(room).emit('
        ↪ blocksuccess', result);
146             }
147             else {
148                 socket.emit('blockfailed',
        ↪ result);
149             }
150         });
151     });
152
153     /*
154     socket untuk membatalkan penawaran
        ↪ spesifik --tested
155     */

```

```

156         socket.on('cancelbid', function(
157             ↪ jsonobject) {
158             var obj = JSON.parse(jsonobject);
159             auctioneerAPI.cancelSpecificBidPrice(
160                 ↪ obj, function(status, result) {
161                     if (status == "1") {
162                         io.to(room).emit('
163                             ↪ cancelbidsuccess',
164                             ↪ result);
165                     }
166                     else {
167                         socket.emit('cancelbidfailed'
168                             ↪ , result);
169                     }
170                 });
171             });
172         });

```

**Kode Sumber A.1** Implementasi Node.js + Socket.io untuk Lelang

```

1
2  var fs = require('fs');
3  var app = require('express')();
4  var options = {
5      key: fs.readFileSync('/etc/nginx/ssl/
6          ↪ lelangapa-socket.key'),
7      cert: fs.readFileSync('/etc/nginx/ssl/
8          ↪ lelangapa-socket.crt')
9  };
10 var httpsserver = require('https').Server(options
11     ↪ , app);
12 var httpserver = require('http').Server(app);
13 var ioServer = require('socket.io');
14 var Promise = require('es6-promise').Promise;
15 var mongo = require('mongodb').MongoClient;
16 var assert = require('assert');

```

```

14
15 var io = new ioServer();
16 var jwt = require('socketio-jwt');
17 require('dotenv').config({path: '/home/grkt/
    ↪ security/.env'});
18
19 app.get('/', function(req, res) {
20     res.status(200).json({
21         status: "success"
22     });
23 });
24
25
26 var url = 'mongodb://localhost:27017/lelangkita';
27
28 function parseRoom(room){
29     var chatInformation = room.split("-");
30     var f = parseInt(chatInformation[0]);
31     var l = parseInt(chatInformation[1]);
32
33     if(f > 1) {
34         //l = f ;
35         //f = parseInt(chatInformation[0]);
36         return {
37             room : chatInformation[1] + '-' +
                ↪ chatInformation[0],
38             from : chatInformation[0],
39             to : chatInformation[1]
40         };
41     }
42     else return {
43         room : room,
44         from : chatInformation[0],
45         to : chatInformation[1]
46     };
47 }
48
49 function insertToRoomCollection(roomCollection,

```

```

    ↪ parsedRoom, updateAt, callback){
50
51  var criteria= {
52    room : parsedRoom.room
53  };
54
55  var insertQuery = {};
56  var fr = parseInt(parsedRoom.from);
57  var to = parseInt(parsedRoom.to);
58
59  if (fr > to) {
60    insertQuery["id_user_1"] = parsedRoom.to;
61    insertQuery["id_user_2"] = parsedRoom.from;
62    insertQuery["room"] = parsedRoom.room;
63  }
64  else {
65    insertQuery["id_user_1"] = parsedRoom.from;
66    insertQuery["id_user_2"] = parsedRoom.to;
67    insertQuery["room"] = parsedRoom.room;
68  }
69  if ( updateAt ) insertQuery["updatedAt"] = new
    ↪ Date();
70
71  //only update created room
72  //why would opening conversation would move the
    ↪ conversation upwards you shitty cunt
73  roomCollection.update( criteria,
74    { $set : insertQuery , $setOnInsert : {
    ↪ createdAt : new Date() } },
75    {upsert:true},
76    function(err, res) {
77      if(callback) callback();
78    });
79
80  }
81  mongo.connect(url, function(err, db) {
82
83    var collection = db.collection('userchat');

```

```

84  var roomcollection = db.collection('chatroom');
85
86  io.use( jwt.authorize({
87
88      secret:process.env.JWT_SECRET,
89      handshake : true
90  }) );
91
92
93  io.on('connection', function(socket){
94
95      socket.emit('handshake',{ status : true });
96
97      socket.on('join-room', function(room) {
98
99          var parsedRoom = parseRoom(room);
100         insertToRoomCollection(roomcollection,
101             ↪ parsedRoom, false, function() {
102             socket.join(parsedRoom.room);
103
104             var cb = function(err, chat) {
105                 if (chat!==[]){
106                     socket.emit('chathistory', chat);
107                 }
108             };
109             collection.find({ room : parsedRoom.room
110                 ↪ }).sort({ sent : -1 }).limit(50).
111                 ↪ toArray(cb);
112
113             });
114
115         });
116
117         socket.on('send', function(data) {
118             var msgParse = JSON.parse(data);
119             var parsedRoom = parseRoom(msgParse.room);
120
121             var insertQuery = {

```

```

119         room : parsedRoom.room,
120         sender : parsedRoom.from,
121         msg : msgParse.body,
122         sent : new Date()
123     };
124
125     collection.insert(insertQuery, function(err
126         ↪ , o) {
127         var ll = parsedRoom.from.toString();
128         if (err) io.to(ll).emit('send-status', {
129             ↪ status : false});
130         else io.to(ll).emit('send-status', {
131             ↪ status : true });
132     });
133
134     //update to inbox
135     insertToRoomCollection(roomcollection,
136         ↪ parsedRoom, true);
137     io.to(parsedRoom.room).emit('new-msg',
138         ↪ insertQuery);
139
140     });
141     });
142
143     httpserver.listen(12000);
144     httpserver.listen(13000);
145     io.attach(httpserver);
146     io.attach(httpserver);

```

**Kode Sumber A.2** Implementasi Node.js + Socket.io untuk *Chat*



## BAB B

### KUISIONER PENGUJIAN *USER EXPERIENCE*



**ITS**  
Institut Teknologi  
Sepuluh Nopember

**KUISIONER TUGAS AKHIR**  
**RANCANG BANGUN APLIKASI LELANG ONLINE**  
**MENGUNAKAN KERANGKA KERJA LARAVEL**  
KI141502 – RONAULI SILVA NS 5113100142  
JURUSAN TEKNIK INFORMATIKA, FAKULTAS TEKNOLOGI INFORMASI ITS

Nama : ROSIDHA HANUM

Umur / Gender / Profesi : 17 / L-P / Pelajar

Intensitas belanja online : 2 kali seminggu / sebulan setahun

Untuk setiap pilihan dibawah, harap melingkari angka sesuai dengan pertimbangan kualitas anda.  
Gunakan skala yang diberikan untuk memilih *quality number* yang sesuai.

Surabaya, Juli 2017


Tandatangan



Aplikaasi Lelang Online Lainnya	Skala					Aplikaasi Lelang Online "Lelangapa"	Skala				
	K	u	r	a	n		K	u	r	a	n
1. Desain & Impresi Web	1	2	3	4	5	2. Desain & Impresi Web	1	2	3	4	5
3. Kejelasan & Konsistensi Sistem	1	2	3	4	5	4. Kejelasan dan Konsistensi Sistem	1	2	3	4	5
5. Kemudahan Penggunaan	1	2	3	4	5	6. Kemudahan Penggunaan	1	2	3	4	5
7. Kejelasan status proses	1	2	3	4	5	8. Kejelasan status proses	1	2	3	4	5
9. Error Message yang jelas	1	2	3	4	5	10. Error Message yang jelas	1	2	3	4	5
11. Performa (kecepatan, reliability)	1	2	3	4	5	12. Performa (kecepatan, reliability)	1	2	3	4	5
13. Rating Keseluruhan	1	2	3	4	5	14. Rating Keseluruhan	1	2	3	4	5
15. Akan merekomendasikan ini pada teman?	1	2	3	4	5	16. Akan merekomendasikan ini pada teman?	1	2	3	4	5

Partisipasi anda sangat membantu kelancaran pengerjaan tugas akhir ini.  
Terimakasih banyak saya sampaikan atas partisipasi anda ☺

**Gambar 2.1** Kuisiонер Pengujian *User Experience* Responden 1



**ITS**  
Institut Teknologi  
Sepuluh Nopember

**KUISIONER TUGAS AKHIR**  
**RANCANG BANGUN APLIKASI LELANG ONLINE**  
**MENGUNAKAN KERANGKA KERJA LARAVEL**  
KI141502 – RONAULI SILVA NS 5113100142  
JURUSAN TEKNIK INFORMATIKA, FAKULTAS TEKNOLOGI INFORMASI ITS

Nama : Cathy M Purba


Umur / Gender / Profesi : 2 / L / Pelajar

Intensitas belanja online : 1 kali seminggu / sebulan / setahun

Untuk setiap pilihan dibawah, harap melingkari angka sesuai dengan pertimbangan kualitas anda.

Gunakan skala yang diberikan untuk memilih *quality number* yang sesuai.

Surabaya, Juli 2017  
Tandatangan




Aplikasi Lelang Online Lainnya	Skala				Aplikasi Lelang Online "Lelangapa"	Skala			
	K	u	r	a		K	u	r	a
1. Desain & Impresi Web	1	2	3	4	2. Desain & Impresi Web	1	2	3	4
3. Kejelasan & Konsistensi Sistem	1	2	3	4	4. Kejelasan dan Konsistensi Sistem	1	2	3	4
5. Kemudahan Penggunaan	1	2	3	4	6. Kemudahan Penggunaan	1	2	3	4
7. Kejelasan status proses	1	2	3	4	8. Kejelasan status proses	1	2	3	4
9. Error Message yang jelas	1	2	3	4	10. Error Message yang jelas	1	2	3	4
11. Performa (kecepatan, reliability)	1	2	3	4	12. Performa (kecepatan, reliability)	1	2	3	4
13. Rating Keseluruhan	1	2	3	4	14. Rating Keseluruhan	1	2	3	4
15. Akan merekomendasikan ini pada teman?	1	2	3	4	16. Akan merekomendasikan ini pada teman?	1	2	3	4

Partisipasi anda sangat membantu kelancaran pengerjaan tugas akhir ini.

Terimakasih banyak saya sampaikan atas partisipasi anda 😊

**Gambar 2.2** Kuisisioner Pengujian *User Experience* Responden 2



**ITS**  
Institut Teknologi  
Sepuluh Nopember

**KUISIONER TUGAS AKHIR**  
**RANCANG BANGUN APLIKASI LELANG ONLINE**  
**MENGUNAKAN KERANGKA KERJA LARAVEL**  
KI141502 – RONAULI SILVA NS 5113100142  
JURUSAN TEKNIK INFORMATIKA, FAKULTAS TEKNOLOGI INFORMASI ITS


Nama : Tatik Rupaiah

Umur / Gender / Profesi : 18 / L-P / Pelajar

Intensitas belanja online : kali seminggu / sebulan / setahun

Untuk setiap pilihan dibawah, harap melingkari angka sesuai dengan pertimbangan kualitas anda.  
Gunakan skala yang diberikan untuk memilih *quality number* yang sesuai.

Surabaya, Juli 2017  
Tandatangan



	Skala						Skala				
	K	u	r	a	n		K	u	r	a	n
<b>Aplikasi Lelang Online Lainnya</b>				Baik	<i>E x c e l l e n t</i>				Baik	<i>E x c e l l e n t</i>	
1. Desain & Impresi Web	1	2	3	4	5	2. Desain & Impresi Web	1	2	3	4	5
3. Kejelasan & Konsistensi Sistem	1	2	3	4	5	4. Kejelasan dan Konsistensi Sistem	1	2	3	4	5
5. Kemudahan Penggunaan	1	2	3	4	5	6. Kemudahan Penggunaan	1	2	3	4	5
7. Kejelasan status proses	1	2	3	4	5	8. Kejelasan status proses	1	2	3	4	5
9. Error Message yang jelas	1	2	3	4	5	10. Error Message yang jelas	1	2	3	4	5
11. Performa (kecepatan, reliability)	1	2	3	4	5	12. Performa (kecepatan, reliability)	1	2	3	4	5
13. Rating Keseluruhan	1	2	3	4	5	14. Rating Keseluruhan	1	2	3	4	5
15. Akan merekomendasikan ini pada teman?	1	2	3	4	5	16. Akan merekomendasikan ini pada teman?	1	2	3	4	5

Partisipasi anda sangat membantu kelancaran pengerjaan tugas akhir ini.  
Terimakasih banyak saya sampaikan atas partisipasi anda ☺

**Gambar 2.3** Kuisisioner Pengujian *User Experience* Responden 3



**ITS**  
Institut Teknologi  
Sepuluh Nopember

**KUISIONER TUGAS AKHIR**  
**RANCANG BANGUN APLIKASI LELANG ONLINE**  
**MENGUNAKAN KERANGKA KERJA LARAVEL**  
K1141502 – RONAULI SILVA NS 5113100142  
JURUSAN TEKNIK INFORMATIKA, FAKULTAS TEKNOLOGI INFORMASI ITS

Nama : Malika

Umur / Gender / Profesi : / L-P / Swasta

Intensitas belanja online : 1 kali seminggu / sebulan / Setahun

Untuk setiap pilihan dibawah, harap melingkari angka sesuai dengan pertimbangan kualitas anda.  
Gunakan skala yang diberikan untuk memilih *quality number* yang sesuai.

Surabaya, 8 Juli 2017

Tandatangan




Malika

Aplikasi Lelang Online Lainnya	Skala					Aplikasi Lelang Online "Lelangapa"	Skala				
	K	u	r	a	n		K	u	r	a	n
1. Desain & Impresi Web	1	2	3	4	5	2. Desain & Impresi Web	1	2	3	4	5
3. Kejelasan & Konsistensi Sistem	1	2	3	4	5	4. Kejelasan dan Konsistensi Sistem	1	2	3	4	5
5. Kemudahan Penggunaan	1	2	3	4	5	6. Kemudahan Penggunaan	1	2	3	4	5
7. Kejelasan status proses	1	2	3	4	5	8. Kejelasan status proses	1	2	3	4	5
9. Error Message yang jelas	1	2	3	4	5	10. Error Message yang jelas	1	2	3	4	5
11. Performa (kecepatan, reliability)	1	2	3	4	5	12. Performa (kecepatan, reliability)	1	2	3	4	5
13. Rating Keseluruhan	1	2	3	4	5	14. Rating Keseluruhan	1	2	3	4	5
15. Akan merekomendasikan ini pada teman?	1	2	3	4	5	16. Akan merekomendasikan ini pada teman?	1	2	3	4	5

Partisipasi anda sangat membantu kelancaran pengerjaan tugas akhir ini.  
Terimakasih banyak saya sampaikan atas partisipasi anda ☺

**Gambar 2.4** Kuisisioner Pengujian *User Experience* Responden 4



**ITS**  
Institut Teknologi  
Sepuluh Nopember

**KUISIONER TUGAS AKHIR**  
**RANCANG BANGUN APLIKASI LELANG ONLINE**  
**MENGUNAKAN KERANGKA KERJA LARAVEL**  
KI141502 – RONALDI SILVA NS 5113100142  
JURUSAN TEKNIK INFORMATIKA, FAKULTAS TEKNOLOGI INFORMASI ITS

Nama : Nur Khoirah

Umur / Gender / Profesi : 40 / L / P /


Intensitas belanja online : 1 kali seminggu / sebulan / setahun

Untuk setiap pilihan dibawah, harap melingkari angka sesuai dengan pertimbangan kualitas anda.

Gunakan skala yang diberikan untuk memilih *quality number* yang sesuai.

Surabaya,      Juli 2017

Tandatangan




Aplikasi Lelang Online Lainnya	Skala					Aplikasi Lelang Online "Lelangapa"	Skala				
	K	u	r	a	E		K	u	r	E	
1. Desain & Impresi Web	1	2	3	4	5	2. Desain & Impresi Web	1	2	3	4	5
3. Kejelasan & Konsistensi Sistem	1	2	3	4	5	4. Kejelasan dan Konsistensi Sistem	1	2	3	4	5
5. Kemudahan Penggunaan	1	2	3	4	5	6. Kemudahan Penggunaan	1	2	3	4	5
7. Kejelasan status proses	1	2	3	4	5	8. Kejelasan status proses	1	2	3	4	5
9. Error Message yang jelas	1	2	3	4	5	10. Error Message yang jelas	1	2	3	4	5
11. Performa (kecepatan, reliability)	1	2	3	4	5	12. Performa (kecepatan, reliability)	1	2	3	4	5
13. Rating Keseluruhan	1	2	3	4	5	14. Rating Keseluruhan	1	2	3	4	5
15. Akan merekomendasikan ini pada teman?	1	2	3	4	5	16. Akan merekomendasikan ini pada teman?	1	2	3	4	5

Partisipasi anda sangat membantu kelancaran pengerjaan tugas akhir ini.  
 Terimakasih banyak saya sampaikan atas partisipasi anda ©

**Gambar 2.5** Kuisisioner Pengujian *User Experience* Responden 5





**ITS**  
Institut Teknologi  
Sepuluh Nopember

**KUISIONER TUGAS AKHIR**  
**RANCANG BANGUN APLIKASI LELANG ONLINE**  
**MENGUNAKAN KERANGKA KERJA LARAVEL**  
KI141502 – RONAULI SILVA NS 5113100142  
JURUSAN TEKNIK INFORMATIKA, FAKULTAS TEKNOLOGI INFORMASI ITS

Nama : KAUMATUL MA'RIFAH


Umur / Gender / Profesi : 20 / L / Pelajar

Intensitas belanja online : kali seminggu / sebulan / setahun

Untuk setiap pilihan dibawah, harap melingkari angka sesuai dengan pertimbangan kualitas anda.  
Gunakan skala yang diberikan untuk memilih quality number yang sesuai.

Surabaya, 8 Juli 2017

Tandatangan



	Skala					Skala				
	K	u	r	a		K	u	r	a	
1. Desain & Impresi Web	1	2	3	4	5	1	2	3	4	5
3. Kejelasan & Konsistensi Sistem	1	2	3	4	5	1	2	3	4	5
5. Kemudahan Penggunaan	1	2	3	4	5	1	2	3	4	5
7. Kejelasan status proses	1	2	3	4	5	1	2	3	4	5
9. Error Message yang jelas	1	2	3	4	5	1	2	3	4	5
11. Performa (kecepatan, reliability)	1	2	3	4	5	1	2	3	4	5
13. Rating Keseluruhan	1	2	3	4	5	1	2	3	4	5
15. Akan merekomendasikan ini pada teman?	1	2	3	4	5	1	2	3	4	5
2. Desain & Impresi Web	1	2	3	4	5	1	2	3	4	5
4. Kejelasan dan Konsistensi Sistem	1	2	3	4	5	1	2	3	4	5
6. Kemudahan Penggunaan	1	2	3	4	5	1	2	3	4	5
8. Kejelasan status proses	1	2	3	4	5	1	2	3	4	5
10. Error Message yang jelas	1	2	3	4	5	1	2	3	4	5
12. Performa (kecepatan, reliability)	1	2	3	4	5	1	2	3	4	5
14. Rating Keseluruhan	1	2	3	4	5	1	2	3	4	5
16. Akan merekomendasikan ini pada teman?	1	2	3	4	5	1	2	3	4	5

Partisipasi anda sangat membantu kelancaran pengerjaan tugas akhir ini.  
Terimakasih banyak saya sampaikan atas partisipasi anda ☺

**Gambar 2.6** Kuisisioner Pengujian *User Experience* Responden 6



**ITS**  
Institut Teknologi  
Sepuluh Nopember

**KUISIONER TUGAS AKHIR**  
**RANCANG BANGUN APLIKASI LELANG ONLINE**  
**MENGUNAKAN KERANGKA KERJA LARAVEL**  
KI141502 – RONALDI SILVA NS 5113100142  
JURUSAN TEKNIK INFORMATIKA, FAKULTAS TEKNOLOGI INFORMASI ITS

Nama : Handy k.N

Umur / Gender / Profesi : 16-P / Pelajar

Intensitas belanja online : kali seminggu / sebulan / setahun

Untuk setiap pilihan dibawah, harap melingkari angka sesuai dengan pertimbangan kualitas anda.  
Gunakan skala yang diberikan untuk memilih *quality number* yang sesuai.


Surabaya, Juli 2017  
Tandatangan



Aplikasi Lelang Online Lainnya	Skala					Aplikasi Lelang Online "Lelangapa"	Skala				
	K	u	r	a	n		K	u	r	a	n
1. Desain & Impresi Web	1	2	3	4	5	2. Desain & Impresi Web	1	2	3	4	5
3. Kejelasan & Konsistensi Sistem	1	2	3	4	5	4. Kejelasan dan Konsistensi Sistem	1	2	3	4	5
5. Kemudahan Penggunaan	1	2	3	4	5	6. Kemudahan Penggunaan	1	2	3	4	5
7. Kejelasan status proses	1	2	3	4	5	8. Kejelasan status proses	1	2	3	4	5
9. Error Message yang jelas	1	2	3	4	5	10. Error Message yang jelas	1	2	3	4	5
11. Performa (kecepatan, reliability)	1	2	3	4	5	12. Performa (kecepatan, reliability)	1	2	3	4	5
13. Rating Keseluruhan	1	2	3	4	5	14. Rating Keseluruhan	1	2	3	4	5
15. Akan merekomendasikan ini pada teman?	1	2	3	4	5	16. Akan merekomendasikan ini pada teman?	1	2	3	4	5

Partisipasi anda sangat membantu kelancaran pengerjaan tugas akhir ini.  
Terimakasih banyak saya sampaikan atas partisipasi anda ☺

**Gambar 2.7** Kuisiонер Pengujian *User Experience* Responden 7



**ITS**  
Institut Teknologi  
Sepuluh Nopember

**KUISIONER TUGAS AKHIR**  
**RANCANG BANGUN APLIKASI LELANG ONLINE**  
**MENGUNAKAN KERANGKA KERJA LARAVEL**  
K1141502 – RONAULI SILVA NS 5113100142  
JURUSAN TEKNIK INFORMATIKA, FAKULTAS TEKNOLOGI INFORMASI ITS

Nama : Gresela Sitorus

Umur / Gender / Profesi : 22 / L - @ / Pelajar


Intensitas belanja online : 3 kali seminggu / sebulan (setahun)

Untuk setiap pilihan dibawah, harap melingkari angka sesuai dengan pertimbangan kualitas anda.

Gunakan skala yang diberikan untuk memilih *quality number* yang sesuai.

Surabaya, 06 Juli 2017

Tandatangan




	Skala					Skala					
	K	u	r	E		K	u	r	E		
Aplikasi Lelang Online Lainnya	a	a	a	x	Aplikasi Lelang Online "Lelangapa"	a	a	a	x		
	n	n	n	c		n	n	n	c		
	g	Ba		e		g	Ba		e		
		ik		l			ik		l		
				l					l		
				e					e		
				n					n		
				t					t		
1. Desain & Impresi Web	1	2	3	4	5	2. Desain & Impresi Web	1	2	3	4	5
3. Kejelasan & Konsistensi Sistem	1	2	3	4	5	4. Kejelasan dan Konsistensi Sistem	1	2	3	4	5
5. Kemudahan Penggunaan	1	2	3	4	5	6. Kemudahan Penggunaan	1	2	3	4	5
7. Kejelasan status proses	1	2	3	4	5	8. Kejelasan status proses	1	2	3	4	5
9. Error Message yang jelas	1	2	3	4	5	10. Error Message yang jelas	1	2	3	4	5
11. Performa (kecepatan, reliability)	1	2	3	4	5	12. Performa (kecepatan, reliability)	1	2	3	4	5
13. Rating Keseluruhan	1	2	3	4	5	14. Rating Keseluruhan	1	2	3	4	5
15. Akan merekomendasikan ini pada teman?	1	2	3	4	5	16. Akan merekomendasikan ini pada teman?	1	2	3	4	5

Partisipasi anda sangat membantu kelancaran pengerjaan tugas akhir ini.  
Terimakasih banyak saya sampaikan atas partisipasi anda ☺

**Gambar 2.8** Kuisisioner Pengujian *User Experience* Responden 8





**ITS**  
Institut Teknologi  
Sepuluh Nopember

**KUISIONER TUGAS AKHIR**  
**RANCANG BANGUN APLIKASI LELANG ONLINE**  
**MENGUNAKAN KERANGKA KERJA LARAVEL**  
K141502 – RONALI SILVA NS 5113100142  
JURUSAN TEKNIK INFORMATIKA, FAKULTAS TEKNOLOGI INFORMASI ITS

Nama : Bunda Pufta

Umur / Gender / Profesi : 33 / L - P /


Intensitas belanja online : 3 kali seminggu sebulan / setahun

Untuk setiap pilihan dibawah, harap melingkari angka sesuai dengan pertimbangan kualitas anda.

Gunakan skala yang diberikan untuk memilih *quality number* yang sesuai.

Surabaya, Juli 2017


Tandatangan



	Skala						Skala														
	K	u	r	a	E		K	u	r	E											
Aplikasi Lelang Online Lainnya	a	n	a	n	g	Baik	Aplikasi Lelang Online "Lelangapa"	a	n	a	n	g	Baik	A	x	c	e	l	e	n	t
1. Desain & Impresi Web	1	2	3	4	5		2. Desain & Impresi Web	1	2	3	4	5									
3. Kejelasan & Konsistensi Sistem	1	2	3	4	5		4. Kejelasan dan Konsistensi Sistem	1	2	3	4	5									
5. Kemudahan Penggunaan	1	2	3	4	5		6. Kemudahan Penggunaan	1	2	3	4	5									
7. Kejelasan status proses	1	2	3	4	5		8. Kejelasan status proses	1	2	3	4	5									
9. Error Message yang jelas	1	2	3	4	5		10. Error Message yang jelas	1	2	3	4	5									
11. Performa (kecepatan, reliability)	1	2	3	4	5		12. Performa (kecepatan, reliability)	1	2	3	4	5									
13. Rating Keseluruhan	1	2	3	4	5		14. Rating Keseluruhan	1	2	3	4	5									
15. Akan merekomendasikan ini pada teman?	1	2	3	4	5		16. Akan merekomendasikan ini pada teman?	1	2	3	4	5									

Partisipasi anda sangat membantu kelancaran pengerjaan tugas akhir ini.  
Terimakasih banyak saya sampaikan atas partisipasi anda ☺

**Gambar 2.9** Kuisisioner Pengujian *User Experience* Responden 9



**ITS**  
Institut Teknologi  
Sepuluh Nopember

**KUISIONER TUGAS AKHIR**  
**RANCANG BANGUN APLIKASI LELANG ONLINE**  
**MENGUNAKAN KERANGKA KERJA LARAVEL**  
KI141502 – RONALDI SILVA NS 5113100142  
JURUSAN TEKNIK INFORMATIKA, FAKULTAS TEKNOLOGI INFORMASI ITS

Nama : Novi Anggun Sapitri

Umur / Gender / Profesi : 1 / L / P

Intensitas belanja online : 3 kali seminggu / sebulan / setahun

Untuk setiap pilihan dibawah, harap melingkari angka sesuai dengan pertimbangan kualitas anda.

Gunakan skala yang diberikan untuk memilih *quality number* yang sesuai.

Surabaya,      Juli 2017

Tandatangan

*Novi*

	Skala						Skala				
	K	u	r	a	n		K	u	r	a	n
1. Desain & Impresi Web	1	2	3	4	5	2. Desain & Impresi Web	1	2	3	4	5
3. Kejelasan & Konsistensi Sistem	1	2	3	4	5	4. Kejelasan dan Konsistensi Sistem	1	2	3	4	5
5. Kemudahan Penggunaan	1	2	3	4	5	6. Kemudahan Penggunaan	1	2	3	4	5
7. Kejelasan status proses	1	2	3	4	5	8. Kejelasan status proses	1	2	3	4	5
9. Error Message yang jelas	1	2	3	4	5	10. Error Message yang jelas	1	2	3	4	5
11. Performa (kecepatan, reliability)	1	2	3	4	5	12. Performa (kecepatan, reliability)	1	2	3	4	5
13. Rating Keseluruhan	1	2	3	4	5	14. Rating Keseluruhan	1	2	3	4	5
15. Akan merekomendasikan ini pada teman?	1	2	3	4	5	16. Akan merekomendasikan ini pada teman?	1	2	3	4	5

Partisipasi anda sangat membantu kelancaran pengerjaan tugas akhir ini.  
Terimakasih banyak saya sampaikan atas partisipasi anda ☺

**Gambar 2.10** Kuisisioner Pengujian *User Experience* Responden 10



**ITS**  
Institut Teknologi  
Sepuluh Nopember

**KUISIONER TUGAS AKHIR**  
**RANCANG BANGUN APLIKASI LELANG ONLINE**  
**MENGUNAKAN KERANGKA KERJA LARAVEL**  
KH141502 – RONALI SILVA NS 5113100142  
JURUSAN TEKNIK INFORMATIKA, FAKULTAS TEKNOLOGI INFORMASI ITS

Nama : FEBRINA CITRA Y

Umur / Gender / Profesi : 18 / L - P / Pelajar

Intensitas belanja online : 1 kali seminggu / sebulan setahun

Untuk setiap pilihan dibawah, harap melingkari angka sesuai dengan pertimbangan kualitas anda.

Gunakan skala yang diberikan untuk memilih *quality number* yang sesuai.

Surabaya, 08 Juli 2017

Tandatangan



Aplikasi Lelang Online Lainnya	Skala					Aplikasi Lelang Online "Lelangapa"	Skala				
	K	u	r	a	n		K	u	r	a	n
1. Desain & Impresi Web	1	2	3	4	5	2. Desain & Impresi Web	1	2	3	4	5
3. Kejelasan & Konsistensi Sistem	1	2	3	4	5	4. Kejelasan dan Konsistensi Sistem	1	2	3	4	5
5. Kemudahan Penggunaan	1	2	3	4	5	6. Kemudahan Penggunaan	1	2	3	4	5
7. Kejelasan status proses	1	2	3	4	5	8. Kejelasan status proses	1	2	3	4	5
9. Error Message yang jelas	1	2	3	4	5	10. Error Message yang jelas	1	2	3	4	5
11. Performa (kecepatan, reliability)	1	2	3	4	5	12. Performa (kecepatan, reliability)	1	2	3	4	5
13. Rating Keseluruhan	1	2	3	4	5	14. Rating Keseluruhan	1	2	3	4	5
15. Akan merekomendasikan ini pada teman?	1	2	3	4	5	16. Akan merekomendasikan ini pada teman?	1	2	3	4	5

Partisipasi anda sangat membantu kelancaran pengerjaan tugas akhir ini.  
Terimakasih banyak saya sampaikan atas partisipasi anda ☺

**Gambar 2.11** Kuisisioner Pengujian *User Experience* Responden 11

*(Halaman ini sengaja dikosongkan)*

## BAB C

### KUISIONER PENGUJIAN *MAINTAINABILITY*

**Tabel 3.1** Kuisioner Pengujian *Maintainability* Responden 1

Pertanyaan	Jawaban
Waktu Pengisian	10 Juli 2017, 10:58
Nama	Stefanus Anggara
Profesi	<i>Mobile Apps Developer</i> di PT. Bukalapak.com
Jumlah proyek/pekerjaan teknis yang pernah dikerjakan	2-5 proyek
Intensitas keterlibatan dalam proyek skala medium-besar dan perubahan yang cepat	0-2 proyek
Tingkat kesulitan proyek/pekerjaan yang pernah dikerjakan	Kebanyakan proyek yg dikerjakan adalah aplikasi <i>mobile</i> , kesulitannya adalah komunikasi / koordinasi dgn pihak <i>backend engineer</i> dan di pihak klien
Penilaian Kode Sumber	
<i>Modularity</i>	5
<i>Self - Descriptiveness</i>	4
<i>Consistency</i>	4
<i>Simplicity</i>	5
<i>Trackability</i>	4
Penilaian Dokumentasi	
<i>Modularity</i>	4
<i>Descriptiveness</i>	4
<i>Consistency</i>	4
<i>Simplicity</i>	3
<i>Trackability</i>	4

**Tabel 3.2** Kuisioner Pengujian *Maintainability* Responden 2

<b>Pertanyaan</b>	<b>Jawaban</b>
Waktu Pengisian	10 Juli 2017, 09:39:20
Nama	M. Luthfie La Roha
Profesi	Pelajar Teknik Informatika ITS
Jumlah proyek/pekerjaan teknis yang pernah dikerjakan	2-5 proyek
Intensitas keterlibatan dalam proyek medium-besar dan perubahan yang cepat	0-2 proyek
Kesulitan proyek/pekerjaan yang pernah dikerjakan	Medium
<b>Penilaian Kode Sumber</b>	
<i>Modularity</i>	4
<i>Self - Descriptiveness</i>	4
<i>Consistency</i>	4
<i>Simplicity</i>	4
<i>Trackability</i>	4
<b>Penilaian Dokumentasi</b>	
<i>Modularity</i>	3
<i>Descriptiveness</i>	3
<i>Consistency</i>	3
<i>Simplicity</i>	3
<i>Trackability</i>	3

**Tabel 3.3** Kuisioner Pengujian *Maintainability* Responden 3

<b>Pertanyaan</b>	<b>Jawaban</b>
Waktu Pengisian	10 Juli 2017, 11:21:28
Nama	Bilflash
Dilanjutkan ke halaman selanjutnya	

Tabel 3.3 – lanjutan dari halaman sebelumnya

<b>Pertanyaan</b>	<b>Jawaban</b>
Profesi	<i>Software Engineer</i> PT Ruma
Jumlah proyek/pekerjaan teknis yang pernah dikerjakan	> 5 proyek
Intensitas keterlibatan dalam pengerjaan proyek skala medium-besar dan fitur perubahan yang sangat cepat	> 2 proyek
Deskripsi tingkat kesulitan proyek/pekerjaan yang pernah dikerjakan	agile
<b>Penilaian Kode Sumber</b>	
<i>Modularity</i>	3
<i>Self - Descriptiveness</i>	2
<i>Consistency</i>	3
<i>Simplicity</i>	3
<i>Trackability</i>	2
<b>Penilaian Dokumentasi</b>	
<i>Modularity</i>	4
<i>Descriptiveness</i>	4
<i>Consistency</i>	4
<i>Simplicity</i>	4
<i>Trackability</i>	4

Tabel 3.4 Kuisioner Pengujian *Maintainability* Responden 4

<b>Pertanyaan</b>	<b>Jawaban</b>
Waktu Pengisian	10 Juli 2017, 21:49:48
Nama	Ardi Nusawan
Dilanjutkan ke halaman selanjutnya	

Tabel 3.4 – lanjutan dari halaman sebelumnya

<b>Pertanyaan</b>	<b>Jawaban</b>
Profesi	Mahasiswa Teknik Informatika ITS
Jumlah proyek/pekerjaan teknis yang pernah dikerjakan	> 5 proyek
Intensitas keterlibatan dalam proyek skala medium-besar dan perubahan yang cepat	> 2 proyek
Kesulitan project yang pernah dikerjakan	Membuat kode yang mudah dimengerti oleh orang lain
<b>Penilaian Kode Sumber</b>	
<i>Modularity</i>	5
<i>Self - Descriptiveness</i>	5
<i>Consistency</i>	5
<i>Simplicity</i>	5
<i>Trackability</i>	5
<b>Penilaian Dokumentasi</b>	
<i>Modularity</i>	5
<i>Descriptiveness</i>	5
<i>Consistency</i>	5
<i>Simplicity</i>	5
<i>Trackability</i>	5

Tabel 3.5 Kuisisioner Pengujian *Maintainability* Responden 5

<b>Pertanyaan</b>	<b>Jawaban</b>
Waktu Pengisian	10 Juli 2017, 22:38:31
Nama	Budi Pangestu Tanujaya
Profesi	Pelajar
Dilanjutkan ke halaman selanjutnya	



Tabel 3.5 – lanjutan dari halaman sebelumnya

<b>Pertanyaan</b>	<b>Jawaban</b>
Jumlah proyek/pekerjaan teknis yang pernah dikerjakan	2-5 proyek
Intensitas keterlibatan dalam proyek skala medium-besar dan perubahan yang cepat	0-2 proyek
Kesulitan project yang pernah dikerjakan	cukup besar - seperti proyek menuntut keamanan dan skalabilitas cukup tinggi ; permintaan yang sering berubah dari fitur awal yang telah didesain
<b>Penilaian Kode Sumber</b>	
<i>Modularity</i>	4
<i>Self - Descriptiveness</i>	5
<i>Consistency</i>	5
<i>Simplicity</i>	3
<i>Trackability</i>	3
<b>Penilaian Dokumentasi</b>	
<i>Modularity</i>	4
<i>Descriptiveness</i>	5
<i>Consistency</i>	5
<i>Simplicity</i>	3
<i>Trackability</i>	3

Tabel 3.6 Kuisioner Pengujian *Maintainability* Responden 6

<b>Pertanyaan</b>	<b>Jawaban</b>
Waktu Pengisian	11 Juli 2017, 19:34:23
Nama	Tri Sutrisno Nusantara
Dilanjutkan ke halaman selanjutnya	

Tabel 3.6 – lanjutan dari halaman sebelumnya

<b>Pertanyaan</b>	<b>Jawaban</b>
Profesi	UX Designer SEVIMA
Jumlah proyek/pekerjaan teknis yang pernah dikerjakan	0-2 proyek
Intensitas keterlibatan dalam proyek skala medium-besar dan perubahan yang cepat	0-2 proyek
Kesulitan project yang pernah dikerjakan	Moderat
<b>Penilaian Kode Sumber</b>	
<i>Modularity</i>	4
<i>Self - Descriptiveness</i>	4
<i>Consistency</i>	3
<i>Simplicity</i>	2
<i>Trackability</i>	2
<b>Penilaian Dokumentasi</b>	
<i>Modularity</i>	3
<i>Descriptiveness</i>	4
<i>Consistency</i>	3
<i>Simplicity</i>	3
<i>Trackability</i>	2

Tabel 3.7 Kuisioner Pengujian *Maintainability* Responden 7

<b>Pertanyaan</b>	<b>Jawaban</b>
Waktu Pengisian	12 Juli 2017, 07:22:02
Nama	Yusro Tsaqova
Profesi	Software Engineer GDP Labs
Jumlah proyek/pekerjaan teknis yang pernah dikerjakan	2-5 proyek
Dilanjutkan ke halaman selanjutnya	

Tabel 3.7 – lanjutan dari halaman sebelumnya

<b>Pertanyaan</b>	<b>Jawaban</b>
Intensitas keterlibatan dalam proyek skala medium-besar dan perubahan yang cepat	> 2 proyek
Kesulitan project yang pernah dikerjakan	Kompleks
<b>Penilaian Kode Sumber</b>	
<i>Modularity</i>	4
<i>Self- Descriptiveness</i>	4
<i>Consistency</i>	3
<i>Simplicity</i>	4
<i>Trackability</i>	4
<b>Penilaian Dokumentasi</b>	
<i>Modularity</i>	4
<i>Descriptiveness</i>	4
<i>Consistency</i>	3
<i>Simplicity</i>	4
<i>Trackability</i>	4

**Tabel 3.8** Kuisioner Pengujian *Maintainability* Responden 8

<b>Pertanyaan</b>	<b>Jawaban</b>
Waktu Pengisian	12 Juli 2017, 08:07:44
Nama	Demsey Iman M
Profesi	Mahasiswa ITS
Jumlah proyek/pekerjaan teknis yang pernah dikerjakan	> 5 proyek
Intensitas keterlibatan dalam proyek skala medium-besar dan perubahan yang cepat	> 2 proyek
Dilanjutkan ke halaman selanjutnya	

Tabel 3.8 – lanjutan dari halaman sebelumnya

<b>Pertanyaan</b>	<b>Jawaban</b>
Kesulitan project yang pernah dikerjakan	proses bisnis yang berubah-ubah dengan cepat / <i>client</i> tidak konsisten
<b>Penilaian Kode Sumber</b>	
<i>Modularity</i>	4
<i>Self - Descriptiveness</i>	5
<i>Consistency</i>	4
<i>Simplicity</i>	4
<i>Trackability</i>	4
<b>Penilaian Dokumentasi</b>	
<i>Modularity</i>	4
<i>Descriptiveness</i>	4
<i>Consistency</i>	4
<i>Simplicity</i>	4
<i>Trackability</i>	4

## BIODATA PENULIS



**Ronauli Silva Natalensis S**, kelahiran & besar di Siantar - Medan, sangat suka belajar. Diberi amanah untuk menjadi *administrator* Laboratorium Pemrograman di tahun 2015, penulis belajar banyak mengenai administrasi *server*, rancang bangun aplikasi terutama di bidang web. Selain itu, beberapa *project* yang diambil penulis mengenai rancang bangun aplikasi yang baik dan buruk yang mengajarkan penulis cara memperbaiki, menangkal dan & mengoptimasinya. Selain itu, penulis juga banyak belajar *softskills* saat diamanahi menjadi sekretaris departemen HMTC ITS 2015/2016 dan juga lewat pelatihan-pelatihan beswan Karya Salemba Empat 2014-2016.

Motto penulis yaitu "*Always go for the extra miles*", membawa penulis mengambil topik tugas akhir ini, dimana penulis dapat menerapkan perbaikan, optimasi dan pelajaran yang penulis petik dari *project-project* sebelumnya, dengan bimbingan dosen-dosen pembimbing penulis yang baik hati. Dalam pendalaman topik tugas akhir ini juga, penulis banyak belajar dan menjadi sangat tertarik mendalami *bussiness engineering, user experiences and usability*, dan *data scientist & engineering*.

Dengan segala kerendahan hati, ilmu penulis masihlah setitik dibandingkan susu sebelanga. Penulis sangat mengharapkan diskusi, ajaran dan bantuan dalam memperbaiki diri. Apabila pembaca berkenan, penulis dapat dihubungi melalui *email* ke [ronayumik@gmail.com](mailto:ronayumik@gmail.com).